US006125197A

United States Patent [19]

Mack et al.

[11] Patent Number: 6,125,197

[45] Date of Patent: Sep. 26, 2000

[54] METHOD AND APPARATUS FOR THE PROCESSING OF STEREOSCOPIC ELECTRONIC IMAGES INTO THREE-DIMENSIONAL COMPUTER MODELS OF REAL-LIFE OBJECTS

[75] Inventors: Walter J. Mack, Chandler; Jean-Charles Korta, Phoenix, both of Ariz.

[73] Assignee: Intel Corporation, Santa Clara, Calif.

[21] Appl. No.: 09/108,274

[22] Filed: Jun. 30, 1998

[51] Int. Cl.^{7} G06K 9/00

[52] U.S. Cl. 382/154; 345/420; 256/12

[58] Field of Search 382/154; 356/12, 356/20, 21, 22; 242/135, 142; 245/419, 420, 425

[56] References Cited

U.S. PATENT DOCUMENTS

5,446,798 8/1995 Morita et al. 382/154

5,818,959 10/1998 Webb et al. 382/154

5,917,937 6/1999 Szeliski et al. 382/154

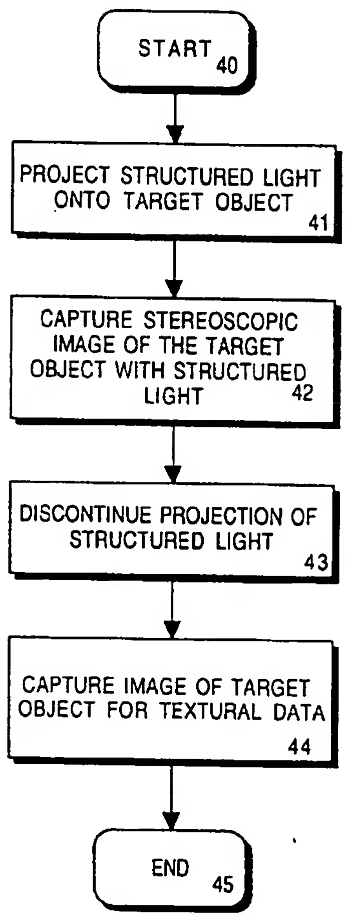
Primary Examiner—Joseph Mancuso

Attorney, Agent, or Firm—Blakely, Sokoloff, Taylor & Zafman LLP

[57] ABSTRACT

A method and apparatus for extracting three-dimensional (3-D) data from a target object is disclosed. A plurality of markers are formed on the object. A plurality of images are captured of the object. A first point is designated from a marker from one of the images and a line equation corresponding to the first point is determined. A second point in a marker in another image corresponding to the first point is determined and a second line equation corresponding to the second point is determined. The intersection of the two line equations is determined.

18 Claims, 12 Drawing Sheets



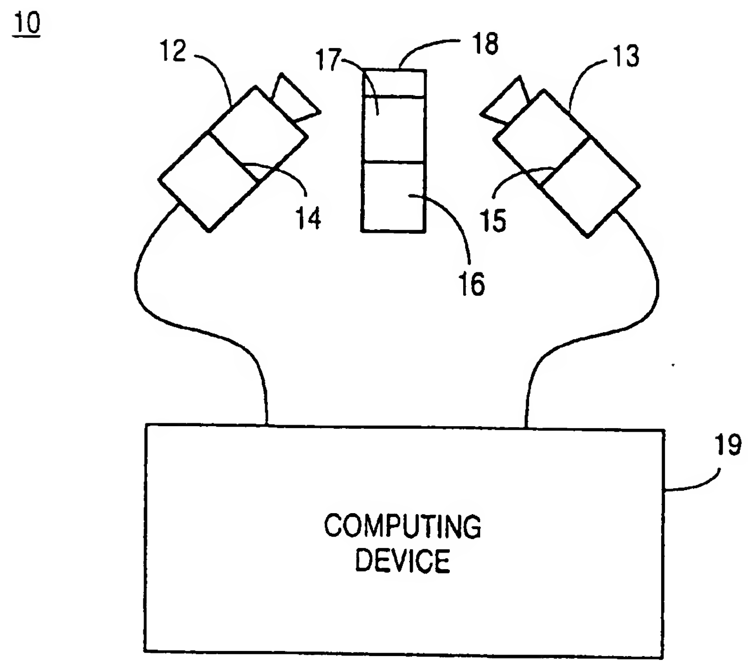

10
12
17
18
13
14
15
16
19
COMPUTING
DEVICE

Fig. 1

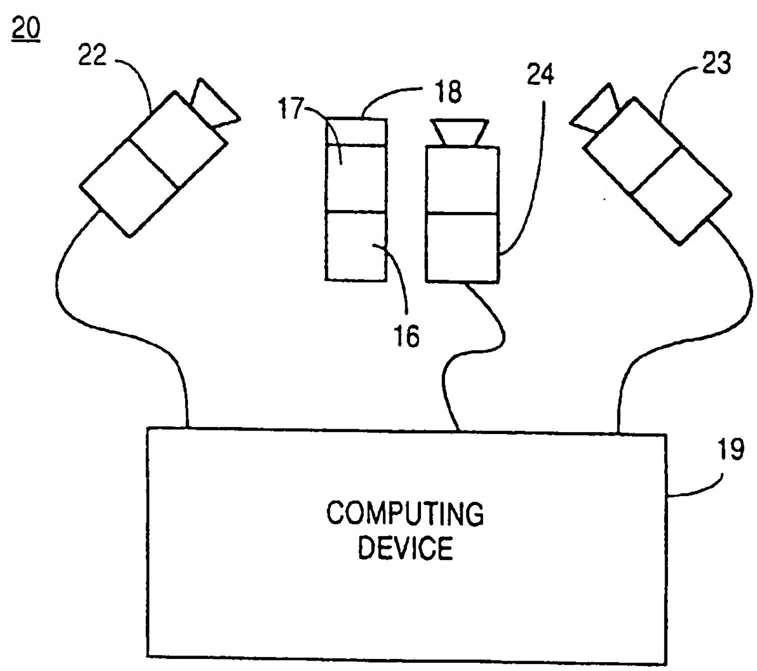

20
22
17
18
24
23
16
19
COMPUTING
DEVICE

Fig. 2

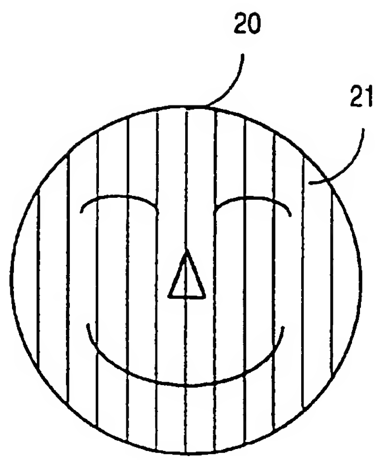


Fig. 3a

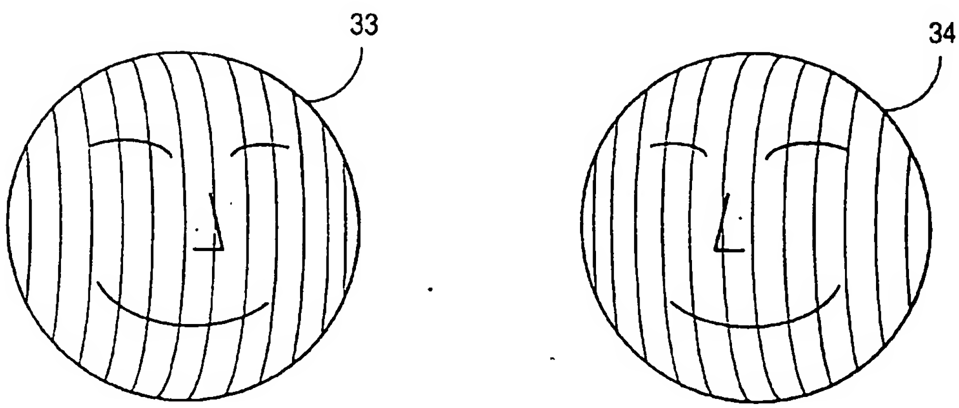


Fig. 3b

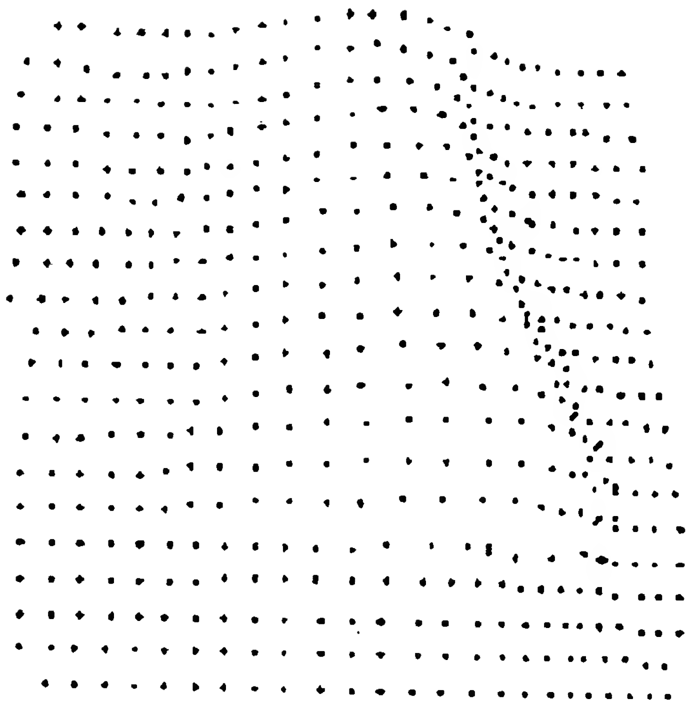


Fig. 3c

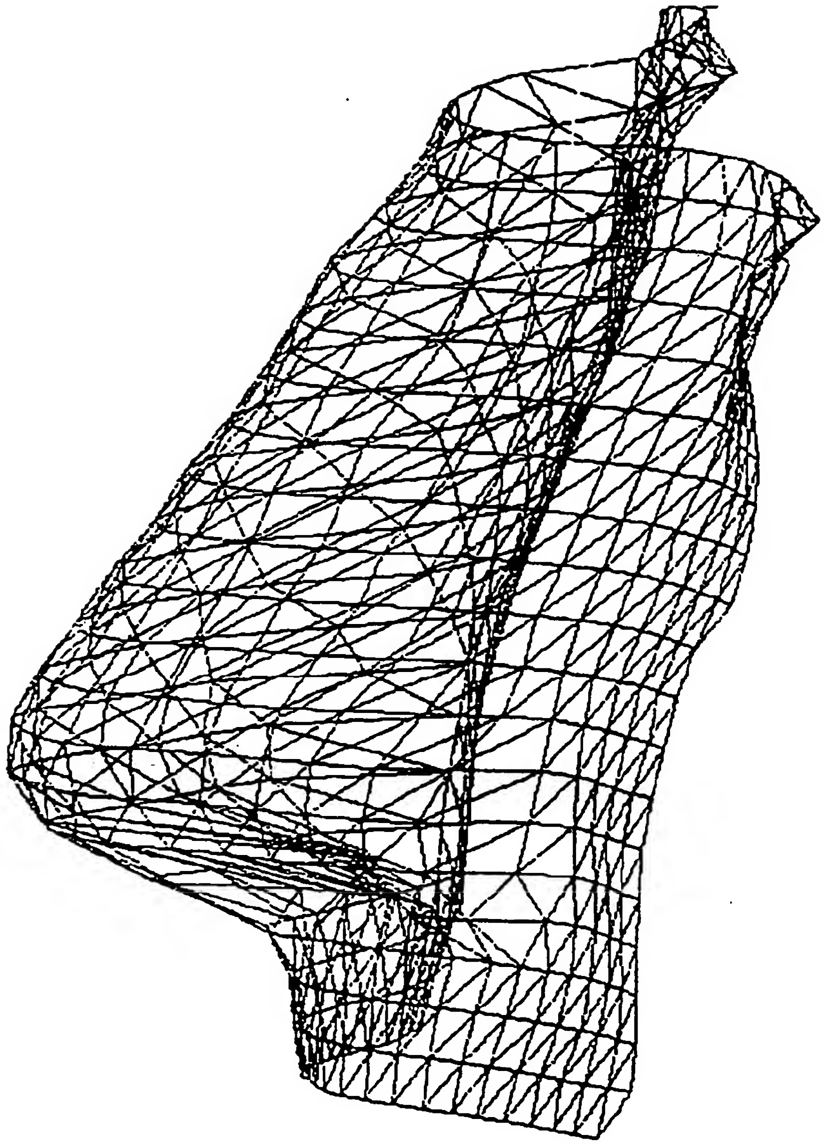


Fig. 3d

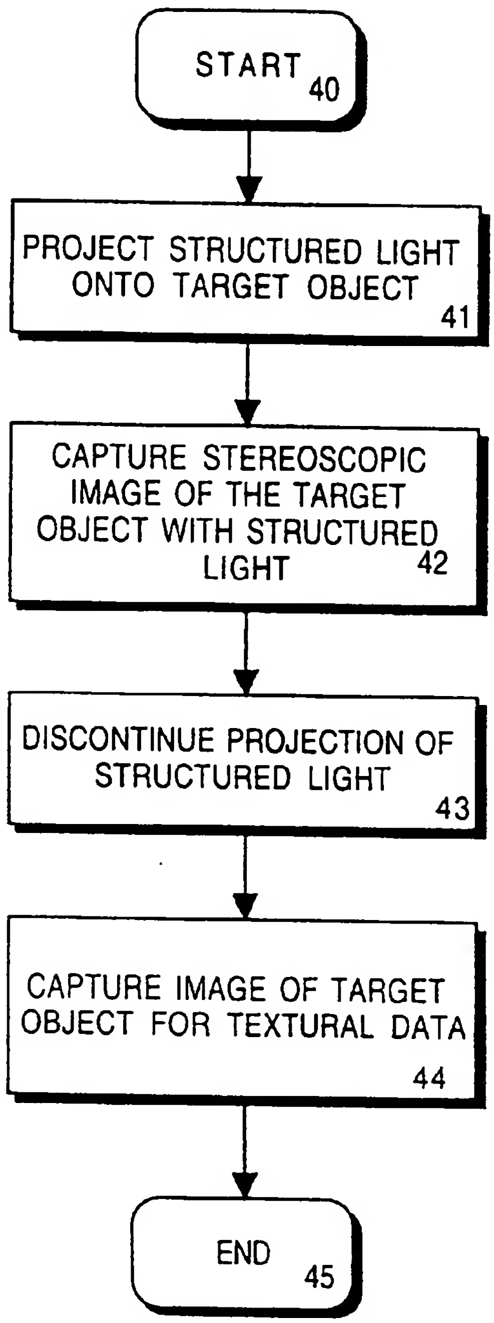

START 40
PROJECT STRUCTURED LIGHT ONTO TARGET OBJECT 41
CAPTURE STEREOSCOPIC IMAGE OF THE TARGET OBJECT WITH STRUCTURED LIGHT 42
DISCONTINUE PROJECTION OF STRUCTURED LIGHT 43
CAPTURE IMAGE OF TARGET OBJECT FOR TEXTURAL DATA 44
END 45

Fig. 4

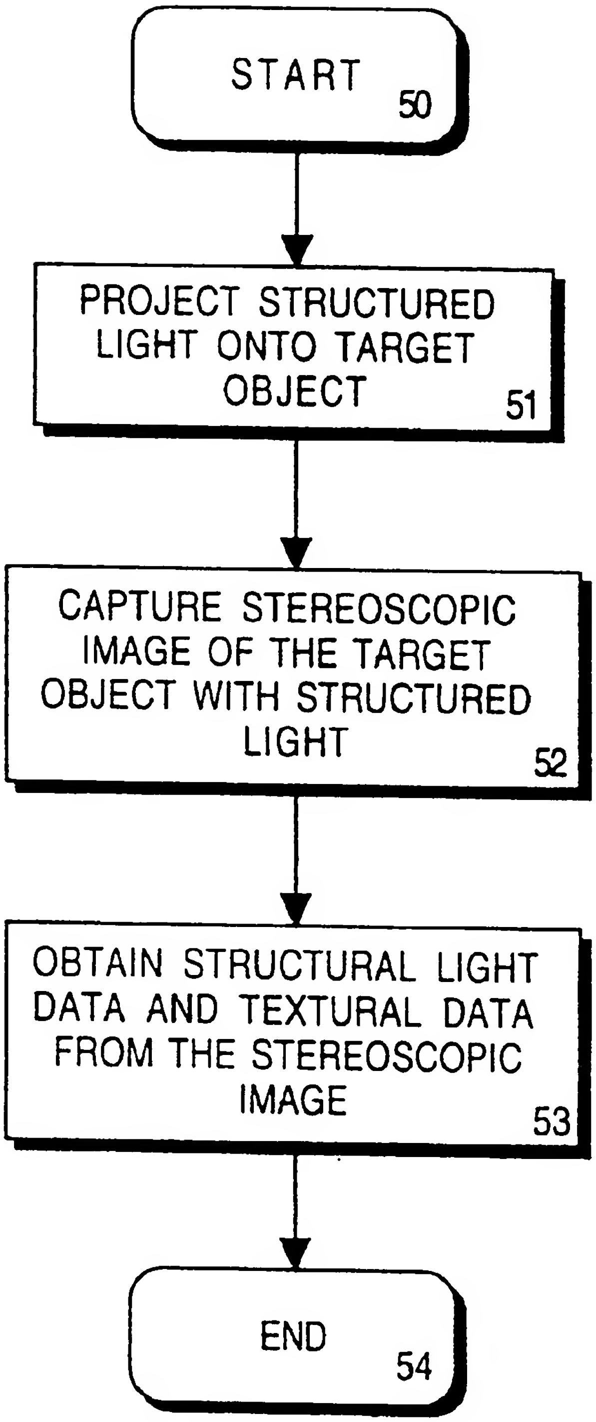

START
50
PROJECT STRUCTURED LIGHT ONTO TARGET OBJECT
51
CAPTURE STEREOSCOPIC IMAGE OF THE TARGET OBJECT WITH STRUCTURED LIGHT
52
OBTAIN STRUCTURAL LIGHT DATA AND TEXTURAL DATA FROM THE STEREOSCOPIC IMAGE
53
END
54

Fig. 5

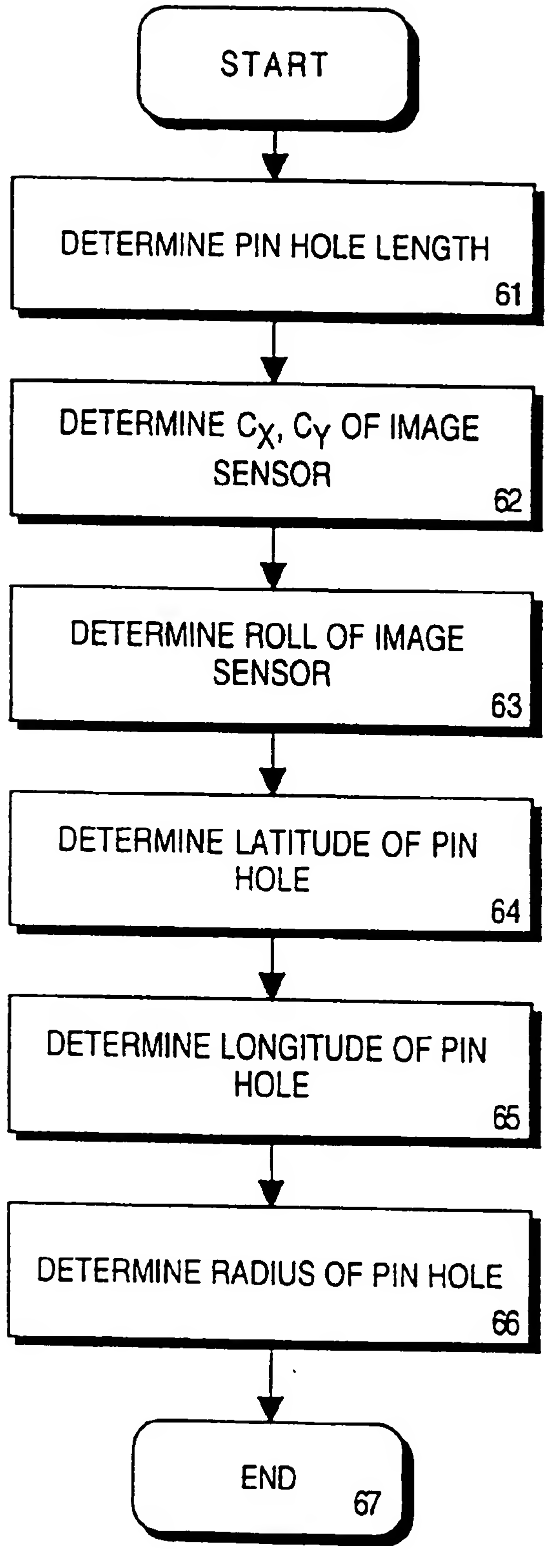

START
DETERMINE PIN HOLE LENGTH
61
DETERMINE C_X, C_Y OF IMAGE SENSOR
62
DETERMINE ROLL OF IMAGE SENSOR
63
DETERMINE LATITUDE OF PIN HOLE
64
DETERMINE LONGITUDE OF PIN HOLE
65
DETERMINE RADIUS OF PIN HOLE
66
END
67

Fig. 6

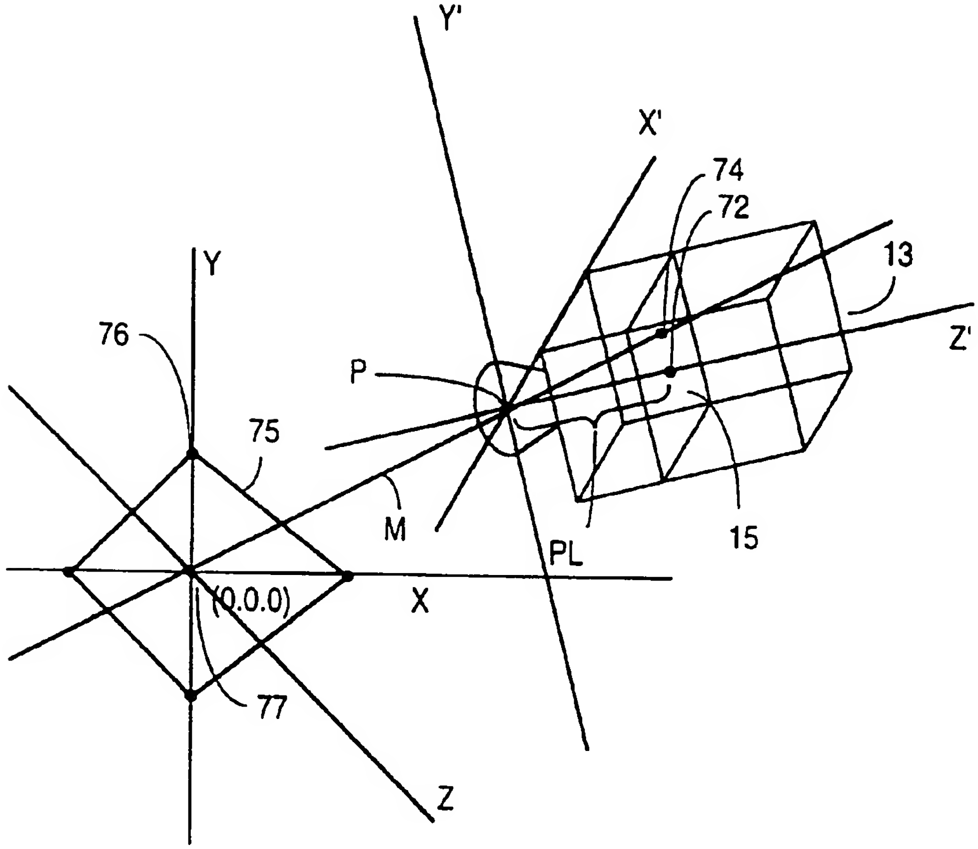


Fig. 7

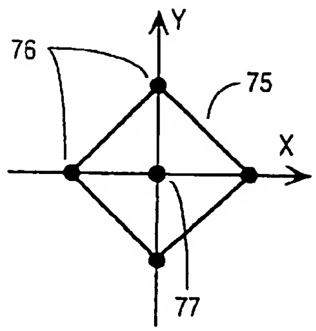


Fig. 8a

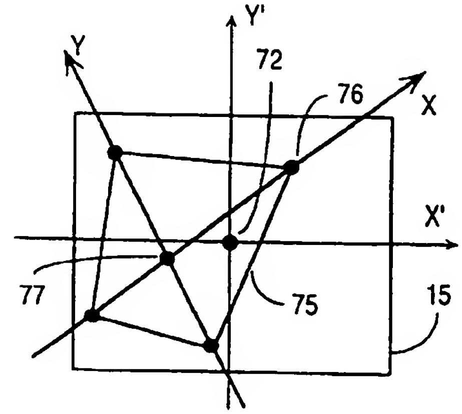


Fig. 8b

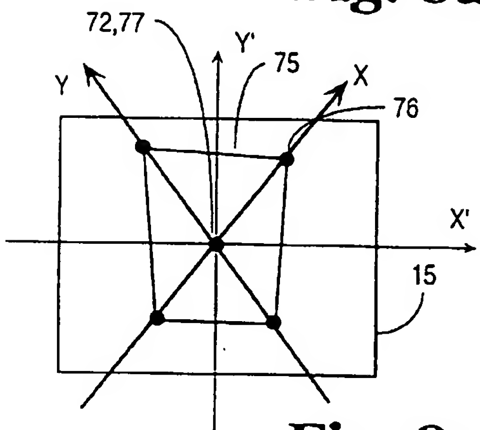


Fig. 8c

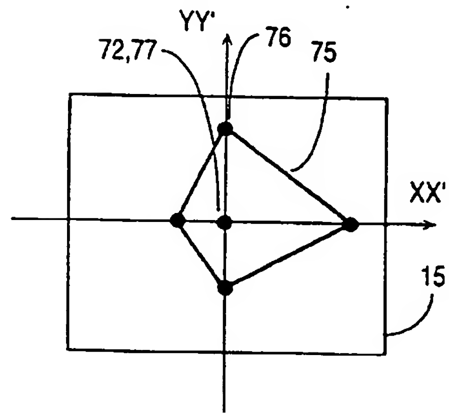


Fig. 8d

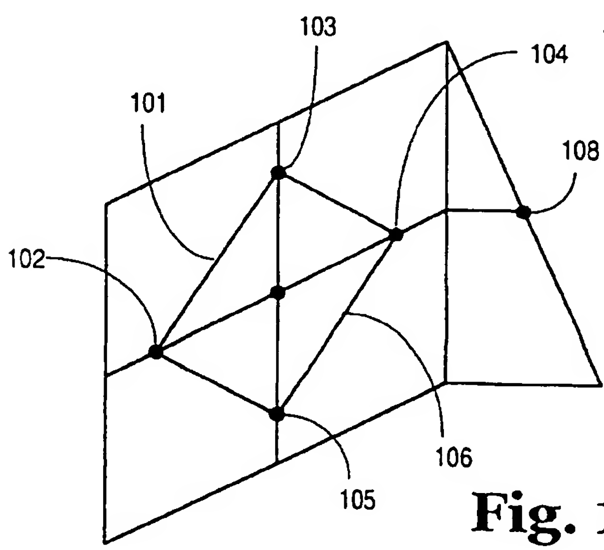


Fig. 10

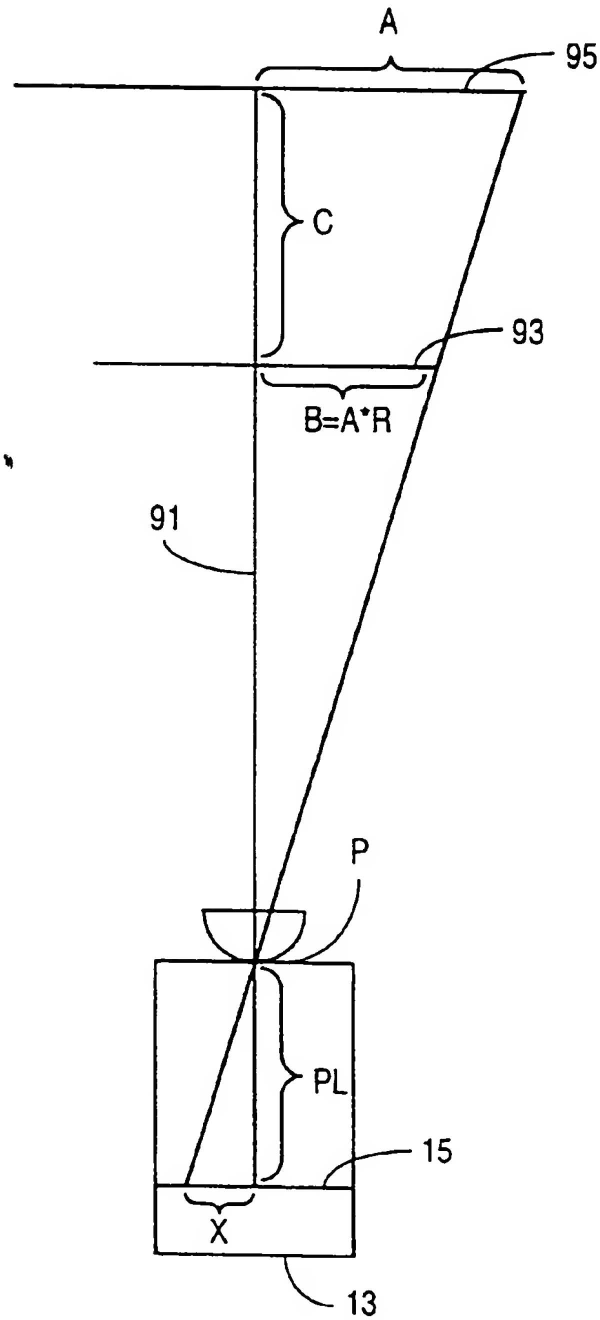


Fig. 9

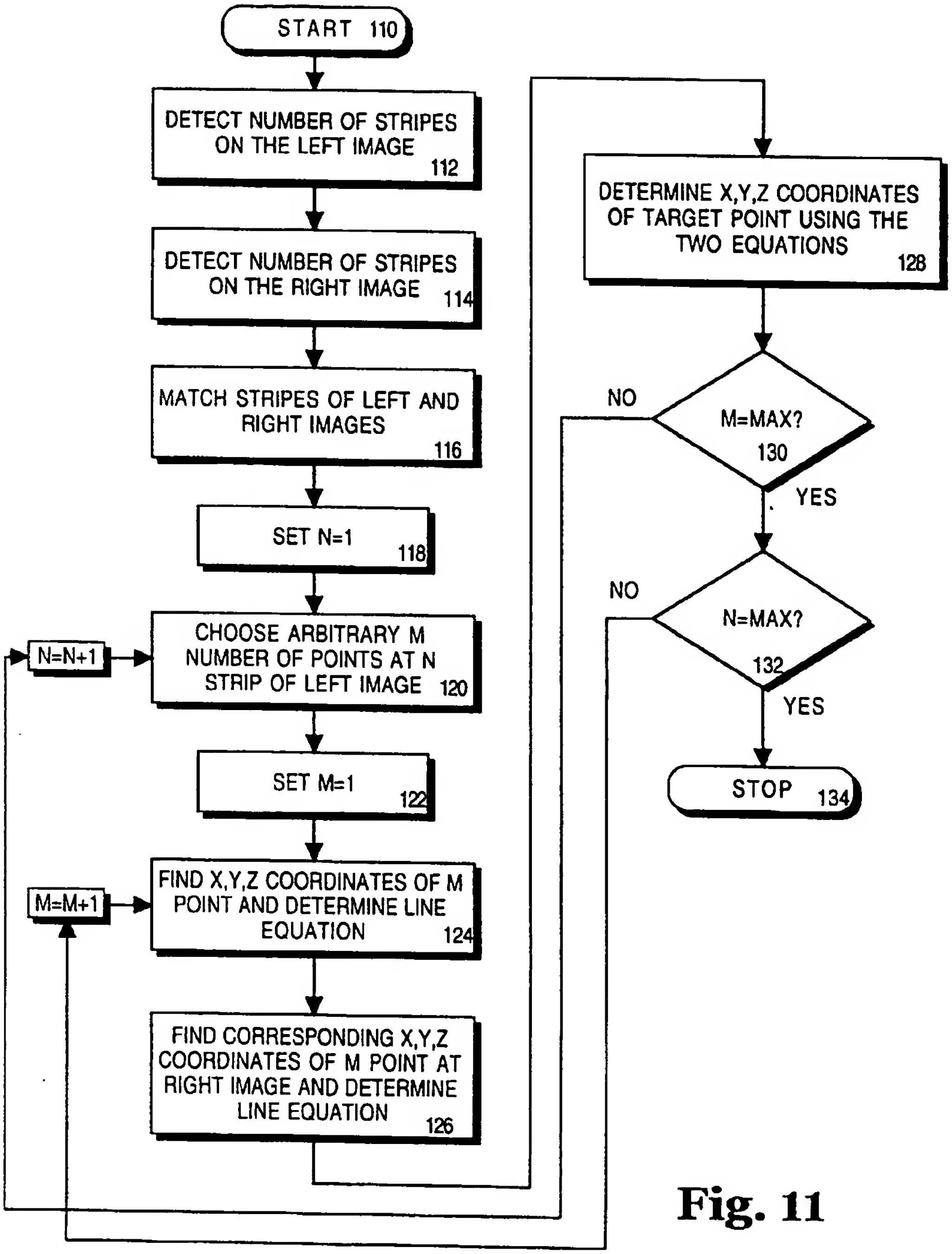

START 110
DETECT NUMBER OF STRIPES ON THE LEFT IMAGE 112
DETECT NUMBER OF STRIPES ON THE RIGHT IMAGE 114
MATCH STRIPES OF LEFT AND RIGHT IMAGES 116
SET N=1 118
N=N+1
CHOOSE ARBITRARY M NUMBER OF POINTS AT N STRIP OF LEFT IMAGE 120
SET M=1 122
M=M+1
FIND X,Y,Z COORDINATES OF M POINT AND DETERMINE LINE EQUATION 124
FIND CORRESPONDING X,Y,Z COORDINATES OF M POINT AT RIGHT IMAGE AND DETERMINE LINE EQUATION 126
DETERMINE X,Y,Z COORDINATES OF TARGET POINT USING THE TWO EQUATIONS 128
NO
M=MAX? 130
YES
NO
N=MAX? 132
YES
STOP 134

Fig. 11

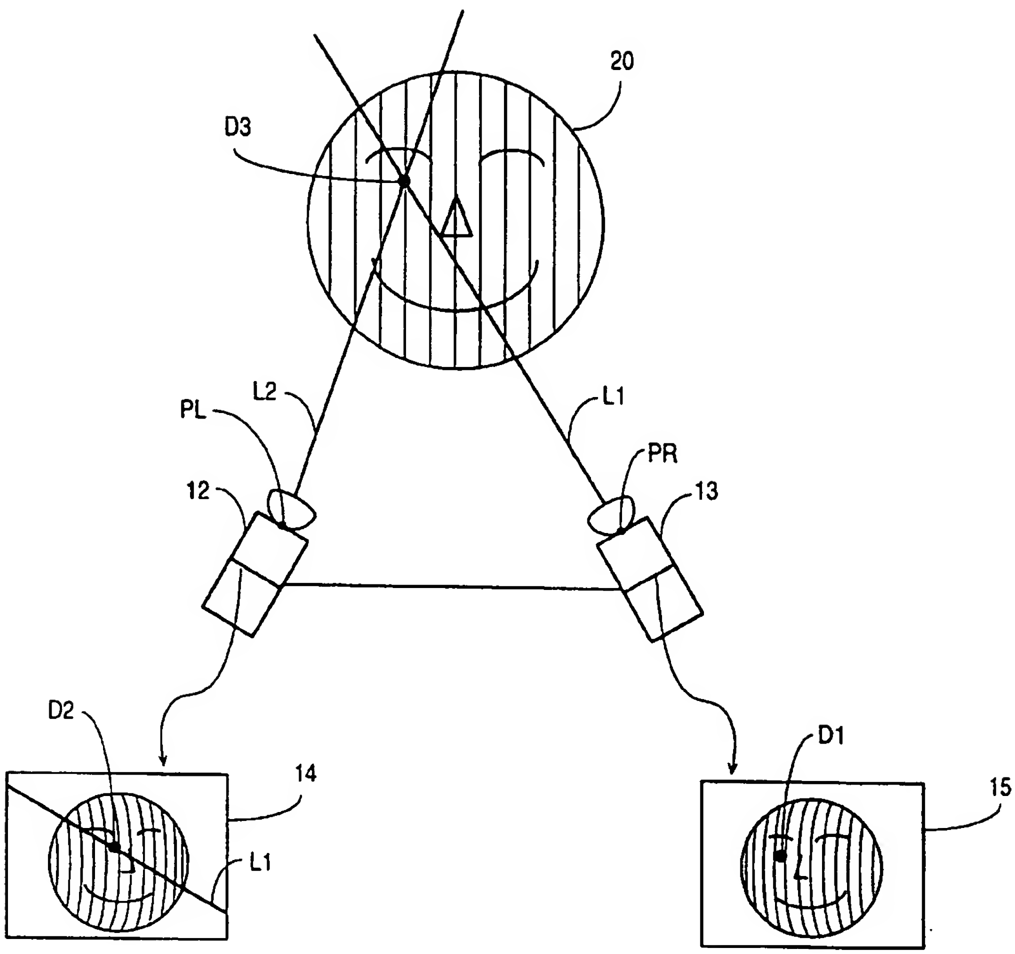


Fig. 12

METHOD AND APPARATUS FOR THE PROCESSING OF STEREOSCOPIC ELECTRONIC IMAGES INTO THREE-DIMENSIONAL COMPUTER MODELS OF REAL-LIFE OBJECTS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to three-dimensional (3-D) models and, in particular, to a method and apparatus for processing stereoscopic images into 3-D data of objects.

2. Description of the Related Art

Creating three-dimensional (3-D) models of objects allow the models to be viewed from many different angles unlike two-dimensional (2-D) models that may only be viewed from one angle. One method of creating 3-D models is to take a multitude of images of real objects from different positions and exploit the differences in the objects' projection. These multitude of images may be suitable to produce 3-D models of the object. Once a 3-D model is produced, the 3-D model could be placed in a virtual world and may be shared with others, much like photos or TV. In particular, the commercial viability of 3-D models is remarkably pronounced in the entertainment, advertisement, and simulation industries.

The desirability and advantages of creating 3-D models of objects are readily understood, and many tools have been developed to produce 3-D models. For example, 3-D software such as 3D Studio MAX from Kinetix, San Francisco, Calif., allows the user to create 3-D models much like an artist would use a lump of clay to sculpture an object, only digitally. As an example, tools such as "metaballs" which are spheres assigned with influence parameters fuse together to form the desired object. For more detailed objects, "polygons" such as triangles or squares defining a small section of the object to be created may be used. Another example would be "splines" which are curves that define a surface area of an object to be created. Details of the creation of 3-D models by software may be found in George Maestri, "Digital Character Animation" (New Riders Publishing, 1996). However, due to the complexity and difficulty of generating 3-D models with software, many have been deterred except those with special skills in the area. For example, the user may need to be artistic as well as technically minded. Further, years of experience may be required before the techniques of 3-D modeling can be mastered. Due to these obstacles, the average consumer, in general, is not able to use the 3-D software to generate 3-D models.

The generation of 3-D data is relatively slow and, as mentioned above, to an average consumer, time consuming and difficult. By using a 3-D imaging device system which can capture objects and subsequently be used to create 3-D data would allow consumers without special expertise to generate 3-D models of real objects expeditiously and with ease. Accordingly, the present invention is directed to a method and apparatus for generating 3-D data using captured stereoscopic images with imaging devices which in turn is used to generate 3-D models of target objects.

BRIEF SUMMARY OF THE INVENTION

A method and apparatus for extracting three-dimensional (3-D) data from a target object is disclosed. A plurality of markers are formed on the object. A plurality of images are captured of the object. A first point is designated from a marker from one of the images and a line equation corresponding to the first point is determined. A second point in a marker in another image corresponding to the first point is determined and a second line equation corresponding to the second point is determined. The intersection of the two line equations is determined.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the method and apparatus for the present invention will be apparent from the following description in which:

FIG. **1** illustrates an exemplary three-dimensional (3-D) imaging device system;

FIG. **2** illustrates another exemplary 3-D imaging device system;

FIG. *3a* illustrates a target object to be stereoscopically imaged,

FIG. *3b* illustrates a stereoscopic image of the target object,

FIG. *3c* illustrates extracted 3-D data, and

FIG. *3d* illustrates surface structure formed on 3-D data;

FIG. **4** illustrates an exemplary method of capturing structural light data of a target object using visible light source and textural data;

FIG. **5** illustrates an exemplary method of capturing structural light data of a target object using non-visible light source and textural data;

FIG. **6** illustrates an exemplary calibration procedure;

FIG. **7** illustrates an imaging device to be calibrated according to a chosen coordinate system;

FIG. *8a* illustrates an exemplary calibration target and

FIGS. *8b–8d* illustrate views of the calibration target from the image sensor's perspective as the image sensor is being calibrated;

FIG. **9** is an exemplary instrument to determine a pin hole length of an imaging device to be calibrated;

FIG. **10** is another exemplary embodiment of a calibration target;

FIG. **11** illustrates an exemplary 3-D data extraction procedure from a stereoscopic image of an object; and

FIG. **12** illustrates another view of an exemplary 3-D data extraction procedure.

DETAILED DESCRIPTION OF THE INVENTION

In creating three-dimensional (3-D) models of real objects, a multitude of images of real objects are taken from different positions to exploit the differences in the objects' projection. The multitude of images may be captured, for example, by an imaging device such as a camcorder or a digital camera comprising an image sensor. The image sensor generally comprises a photosensitive pixel array where each pixel captures an incident light falling on it. Thus, the combination of the pixels within the pixel array is able to capture an image from the incident light. Generally, a surface of the object to be re-constructed into a 3-D model should be present in at least two images since the re-construction is based on the surface intersection of the two images. These two or more images (i.e. stereoscopic image) of the object having the above characteristics are suitable for subsequent processing into 3-D models.

In taking a stereoscopic image of an object, several issues are taken into consideration. First, the object should not

change its position or shape while the images are taken. Second, the object should be such that the features on its surface are located identically for all images. This is not true for specular reflections where the features change locations as the location of the imaging device is changed. Third, the path of the light should be essentially straight for the invention to function properly. In other words, the projection of an image should not be distorted due to modifications in its path. As an example, glass should not be in the path of the projected image since glass has the tendency to bend a path of light. However, one skilled in the art will recognize that an image may be compensated for the discrepancies if the mentioned conditions exist during stereoscopic image capture.

FIGS. 3a–3c illustrate an example of 3-D data being extracted from a target object. FIG. 3a is the target object, in this instance, a face. By capturing images of the object from different positions using an imaging device, face 33 and face 34 may be captured as illustrated in FIG. 3b which is a stereoscopic image of the face. One could verify that the projection of the face as captured by the imaging device changes predictably as the position of the imaging device is changed. Note the images captured by the imaging device contain only two-dimensional (2-D) data because each image is contained in the plane of the image sensor which is two dimensional. However, where there is an intersection of at least two images of a surface of the face from two different positions, a depth Z value may be obtained for each point in one image that has a corresponding point in the other image with respect to an arbitrary coordinate system having X, Y, Z coordinates that is common to both image sensors. Furthermore, the positions of the two corresponding points in the two image sensors expressed by X', Y' coordinate in the image plane may be combined to produce a 3-D point (i.e. X, Y, Z value) which may be one of the points in 3-D data. Thus, 3-D data is a plurality of points in 3-D space identified within a coordinate system to form a 3-D point image of the target object, an example which is illustrated in FIG. 3c. Obtaining X, Y, Z value of a point in the target object will be described further below. Examples of 3-D imaging device systems to capture stereoscopic images of objects will now be described.

FIG. 1 illustrates an embodiment of a 3-D imaging device system. The 3-D imaging device system 10 illustrated is shown with two imaging devices which for ease of understanding the invention will be designated left imaging device 12 and right imaging device 13. The designation is mentioned throughout the description, however, one skilled in the art would recognize from reading the description that the designation may be interchangeable and further, the invention is applicable where more than two imaging devices are used or under suitable conditions, where only one imaging device is used. Each imaging device 12 and 13 comprises an image sensor 14 and 15 that is able to capture an image of a target object. The 3-D imaging device system 10 may include a computing device 19 to process a stereoscopic image captured by the imaging devices 12 and 13 into 3-D data. The computing device may be a microprocessor, an arithmetic logic unit (ALU) or any other devices capable of processing data information. In one embodiment, the computing device 19 may even process the 3-D data into 3-D models depending on the sophistication of the underlying software. As an example, 3-D data may be "triangulated" (i.e. forming the surface of the object by forming triangles with every three points of the 3-D data) using conventional algorithm such as Delaunay's algorithm. One skilled in the art will recognize that other algorithms may be used including suitable geometric structures. An example of a triangulated structure is illustrated in FIG. 3d. Textural data may be applied to the triangulated structure by using, for example, True Space, a software commercially available from Caligary, Mountain View, Calif. Generally, textural data comprises material information such as physical surface properties of an object and may also comprise color information of the object. Alternatively, the images may be stored in the imaging devices 12 and 13 to be processed at a later time eliminating the need for a computing device in the 3-D imaging device system. Generally, "calibration" information, to be described further below, relating to the 3-D imaging device system 10 is stored in a memory device which may be coupled with or may be part of the computing device 19. However, in a 3-D imaging system 10 where the computing device 19 is not used, the system 10 may comprise a memory device to store calibration information or the calibration information may be separate from the system and introduced when the stereoscopic images are being converted into 3-D data.

The 3-D imaging device system 10 may further comprise a light device 16 to project an originating light beam and a diffracting device 17 to split the beam into an adequate pattern of lines, grids, dots or any other geometrical patterns. As an example, the diffracting device may be one commercially available from Digital Optics Corporation, Charlotte, N.C. or Rochester Photonics, Rochester, N.Y. The term "structured light" will be understood to mean structures comprising lines, strips, segmented lines, grids, dots, etc. produced by a light source. The reason for the structured light is to provide a structure to the target object that is easily recognizable by a computing device. In other words, it is difficult for the computing device to match one point in one image to the corresponding point in another image with information obtained from the natural features of the target object alone. As an example, if a human face is the object to be 3-D modeled and the eye is the feature to be matched, the computing device may err because it may not be able to distinguish between the two eyes in the other corresponding image or the other features. However, by using structured light, the contours of the object can be easily referenced by a computing device in terms of the location of the vertical lines, for example. In one embodiment, each vertical line in the structured light may be coded to distinguish one vertical line from another. Coding is desirable where the contour of a target object causes the vertical lines to emerge, for example. In this instance, the computing device may err by jumping from one vertical line to another. By providing a code for each line, the computing device knowing the code of the vertical line will detect an error when the code of the line being traced has changed and may be able to re-trace back to correct the error. The lines may be coded by using a suitable filter 18, for example, coupled with one end of the diffracting device 17. For example, the code may be the segmentation of the vertical line into a pattern different from the other vertical lines in the close proximity. The filter 18 may have slits allowing the vertical lined structured light to be emitted but may have blockages in the slits corresponding to the desired segmented patterns of vertical lines.

SPLIT

STRUCTURED LIGHT

One embodiment of the light source 16 may be a laser emitter. The light source 16 and the diffracting device 17 may project a laser light, for example, vertical lines onto the target object. The description of the vertical lines should by no means be interpreted as limiting the scope of the invention. For example, horizontal lines may be used depending on the orientation of the cameras. Furthermore, it may be a grid of dots or segmented lines, etc. FIG. 4 illustrates one

IMAGE CAPTURE SEQUENCE

example of capturing a stereoscopic image of a target object using visible light source, and textural data. Block 41 illustrates a structured light projected onto the target object by a light source 16. With the structured light projected, block 42 illustrates a stereoscopic image of the object captured by the left 12 and right imaging devices 13 of the 3-D imaging device system 10. Once the stereoscopic image of the object with a structured light (i.e. structural light data) is captured, block 43 illustrates the light source 16 is switched off. Block 44 illustrates simultaneously or as close to simultaneously as possible, an image is captured by one of the left imaging device 12 and right imaging device 13 to obtain the textural data. Although multiple images may be taken by the left 12 and right imaging devices 13 to obtain textural data, generally, a single image from one imaging device may be sufficient. The reason for the simultaneousness is to match as closely as possible the stereoscopic image with the structural light data to the image with textural data. It should, however, be noted that where the 3-D imaging device system 10 and the target object are relatively stationary, simultaneousness is no longer important. Furthermore, in one embodiment, the textural data may be independent to the stereoscopic image, for example, where the textural data may be computer generated using conventional methods, textural image is not required.

WAVELENGTH

In another embodiment, light source 16 may emit light in the infra-red region (generally considered to be light with wavelength longer than 780 nm). In this instance, the image sensor 14, 15 of the imaging device 12, 13 may be designed to enable simultaneous capture of the visible light textural data and infra-red structured light data. The image sensors 14, 15 may be equipped to simultaneously capture visible and infra-red light through appropriate use of color filters. As an example, an image sensor comprising a 2x2 square pattern of red, green, blue (RGB) and infra-red (IR) pixels may be created using existing commercial Color Filter Array (CFA) materials, taking advantage of the fact that these materials are transparent to IR radiation. By a simple overlay of two CFA colors (e.g. R, B) that have no overlapping transmittance in the visible portion of the spectrum, it is possible to create a composite filter element which blocks the visible light and transmits only IR. If two filters are used to form the composite filter, then each of the two filters has a visible radiation pass spectrum that is disjoint from the other, so that there is substantially no transmittance of visible light through the resulting composite filter formed from the combination of the two filters. If more than two filters are used, then each filter has a visible radiation pass spectrum such that the resulting composite filter is substantially opaque to visible light. This composite filter element is thus an IR pass filter, because each of the component filters used to form the composite filter is substantially transparent to IR. The deposition of the CFAs are accomplished by photo-lithographic techniques well known to the semiconductor industry. Further information on RGB and IR image sensor may be found in a pending application titled "Infrared Pixel Sensor and Infrared Signal Correction", Ser. No. 75/041,976, filed on Mar. 13, 1998.

FIG. 5 illustrates one example of capturing structural light data of a target object using non-visible light source, and textural data. Block 51 illustrates the non-visible light source projecting a structured light onto the target object. Block 52 illustrates a stereoscopic image taken by the left imaging device 12 and right imaging device 13. Block 53 illustrates at least one of the imaging devices 12, 13 generating color outputs (e.g. red, blue, green) for textural data and both imaging devices 12, 13 generating non-visible light output (e.g. infra-red) for processing of structural light data.

In another embodiment, the imaging device system 10 comprises a combination of monochrome imaging devices such as black and white imaging devices and color imaging devices. Generally, where color imaging devices are used to capture both the textural data and the structured light data of the target object, the resolution of the structured light data is compromised. This arises from color generally being defined by three or more pixels (e.g. red, green and blue) that is also used to define a point in the feature, hence, positional information is spread over a group of pixels. By using one pixel to define one point position in the feature, higher resolution may be obtained. In using monochrome imaging devices to capture positional information of the features, a higher degree of accuracy may be achieved.

FIG. 2 illustrates an embodiment of the 3-D imaging device system described above. The 3-D imaging device system 20 may comprise a pair of monochrome imaging devices designated here as a left monochrome imaging device 22 and right monochrome imaging device 23. The pair of monochrome imaging devices 22, 23 capture a stereoscopic image of a target object that comprises structural light data of the object. The structured light is produced by a light source 16 and a diffracting device 17 in a manner described previously. Where coding of the structured light is desired, an appropriate filter 18 may be coupled with the diffracting device 17. The 3-D imaging device system 20 further comprises a color imaging device 24 that captures the textural data of the target object. When textural data is to be obtained, the light source 16 is prevented from emitting structured light if the light is in the visible spectrum. However, if the light is in the non-visible spectrum, the monochrome imaging devices 22, 23 and the color imaging device 24 may take an image of the target object simultaneously. In this instance the monochrome imaging devices should be adaptable to capture structural light data produced by the light in the non-visible spectrum. As an example, where the non-visible light source is an infra-red light emitter as described previously, the monochrome imaging devices may be an infra-red imaging device. It should be noted that all three imaging devices 22, 23, 24 should be calibrated with respect to a chosen coordinate system such that the captured positional information obtained by the monochrome imaging devices 22, 23 may be combined with the textural data of the color imaging device 24 to form a 3-D model with high degree of resolution. Before any stereoscopic images are taken, all imaging devices in a 3-D imaging device system should be calibrated which will be apparent with the description below.

Referring to FIG. 1 as an example, calibration is performed on imaging devices 12 and 13 to determine each position and orientation of the imaging devices before any stereoscopic images are captured. By performing calibration, the imaging devices are placed in a chosen coordinate system to be described further below that allows the computing device used to create the 3-D data to know the relative position of the imaging devices in the chosen coordinate system. With the position of the imaging devices known, features of the captured stereoscopic images may be correlated together to form a combined input in order to form the 3-D data for 3-D modeling. To illustrate this point, imagine two imaging devices in 3-D space taking an image of the same object to form a left image and a right image of the object which is the stereoscopic image of the object. Due to the differences in the two images, stereoscopic matching can take place. Stereoscopic matching is a process where a point feature in one image is matched with the corresponding point feature in the other image. While the human visual

system can readily detect the various features of the left image and the right image, and correlate the two images together, a computing device performing a similar function would need to define the various features in terms of coordinates in a coordinate system. The relevant information from this activity is the set of coordinates for each image which determines the location of the features in the image. The coordinate set of these features in all images, together with the position of the imaging devices with which each image was taken can then be used to determine the original location in 3-D space of the identified feature.

FIG. 6 illustrates one embodiment of performing calibration. For calibration purposes, a total of at least six positional values may be required for a complete description of the position and orientation of an imaging device with respect to a chosen coordinate system. It should be noted that the positional values are determined for each imaging device in the 3-D imaging device system. In one embodiment, the imaging device may be defined as an image sensor **15** with a pin hole P projecting from the center and normal to the image sensor **15** at a predetermined length (i.e. pin hole length to be described further below) as illustrated in FIG. 7. A pin hole of an imaging device is a fictitious point in space located a fixed distance from and normal to the center of the image sensor where all the incident light corresponding to an image enters the imaging device to project the image on the image sensor. The position of the imaging device may be determined by the position of the pin hole in the chosen coordinate system. In one embodiment, the chosen coordinate system may be a cartesian coordinate system with the origin and X, Y, Z, axis designated arbitrarily, thus, three of the positional values may be X, Y, Z, corresponding to the position of the pin hole in the chosen coordinate system. In another embodiment, polar coordinate system may be used and similarly the origin and radius, latitude angle, longitude angle reference are designated arbitrarily, thus, the position of the pin hole may be defined in radius, longitude angle, and latitude angle in the chosen coordinate system.

The orientation of the imaging device may be determined by the orientation of the image sensor with respect to the chosen coordinate system. In determining the orientation of the image sensor, each imaging device may be designated an imaging device coordinate system. For example, the origin of the imaging device coordinate system may be the pin hole of the imaging device. The Z'-axis of the imaging device coordinate system may be the axis passing through the pin hole and the center of the image sensor. The X'-axis and the Y'-axis of the imaging device coordinate system may be parallel to a horizontal and vertical side of the image sensor respectively. One skilled in the art will recognize that different origin and orientation of the axis may be used for the imaging device coordinate system. In one embodiment, a polar coordinate system may be used where initially, an imaging device to be calibrated may be placed in an orientation in the chosen coordinate system where the pin hole may lie at the origin of the chosen coordinate system and the center of the image sensor may lie at the Z-axis of the chosen coordinate system, the image sensor intersecting the Z-axis at a distance of a pin hole length. The X'-axis and the Y'-axis of the image sensor may be parallel with the X-axis and Y-axis of the chosen coordinate system respectively. When the pin hole is moved in radius, longitude angle, latitude angle to its actual position in the chosen coordinate system, the image sensor would also move from its initial orientation to a known orientation designated as reference orientation in the chosen coordinate system. The actual orientation of the image sensor may be measured as a deviation from the reference orientation. In one embodiment, the deviation may be determined through the remaining three positional values that correspond to Cx, Cy and roll which will be described with more detail further below. Note that the orientation is such that the image plane center is on the axis formed by the origin of the chosen coordinate system and the pin hole location as described above.

In one embodiment, a calibration target may be used to determine the position and orientation of the imaging device in a chosen coordinate system. For illustration purposes, the polar coordinate system is used. An exemplary calibration target **75** comprising a diamond represented by a dot on each corner **76** and the center **77**, totaling five dots as illustrated in FIG. 8*a* may be used. However, it will be apparent to one skilled in the art that other configurations and shapes may be used to achieve a desired result. Note that a calibration target may be a drawing or sets of points on a piece of paper or it may be an actual object. If an actual object is used, the object should have features that may be used as reference points. As an example, a face of a cube may be used as a calibration object using the corners of the face as reference points with perhaps, another reference point defined at the center by the intersection of two diagonal imaginary lines, each line connecting two corners of the face.

Referring back to FIG. 7, the coordinate of the pin hole P of the imaging device **13** may be defined according to a chosen coordinate system the origin of which, for example, may be the center of the calibration target **75** and having a X-axis that may be horizontal to the calibration target, Y-axis that may be vertical to the calibration target, and Z-axis that may be normal to the calibration target as illustrated. It should noted that the chosen coordinate system should be the same for all imaging devices to be calibrated in a 3-D imaging device system so that each imaging device would have a common chosen coordinate system. The coordinate of the pin hole P may be defined in Radius, Latitude angle, Longitude angle, corresponding to three positional values with respect to the center dot **77** of the calibration target **75**. Radius, latitude angle, and longitude angle can readily be produced from the description of the position in the cartesian coordinate system defined by the X, Y, and Z axis which is conventional.

The orientation of the image sensor **15** may be determined by its center **72** with respect to an axis M defined by the center dot **77** of the calibration target **75** and the coordinate of the pin hole P. The center dot **77** of the calibration target **75** which lies on the axis M will be imaged at the image sensor **15** representing the deviation **74** from the center **72** of the image sensor **15** with respect to the axis M. From the deviation **74**, the center **72** of the image sensor **15** may be aligned with the axis M by rotating the image sensor **15** about the X'-axis (Cx) and the Y'-axis (Cy) with respect to the pin hole P in the imaging device coordinate system described above until center **72** corresponds to the location of deviation **74**. The angular values of Cx and Cy corresponds to two of the remaining positional values. The roll of the image sensor **15** defines the rotational orientation of the image sensor **15** in the Z'-axis in the imaging device coordinate system. Roll is compensated for by rotating the image sensor **15** along the Z'-axis until the Y-axis of the coordinate is parallel with the Y'-axis of the image sensor from the image sensor's perspective, for example. The angular value of the roll corresponds to the last remaining positional value.

It should be noted that the exemplary six positional values may be dictated according to the sequence in which the

values are applied. In other words, positional values for one sequence may not be interchangeable with positional values for another sequence. For example, positional values obtained for the sequence: Cx, Cy, roll, latitude, longitude, radius may be different from positional values obtained for the sequence: roll, Cx, Cy, longitude, latitude, radius. Hence, positional values are identified with the sequence of the positional values taken.

Referring back to FIG. 6, it should be noted that the sequence illustrated should by no means be construed as a limitation and one skilled in the art will recognize that other sequences may be used. Block 61 illustrates determining the pin hole length which is the length distance of the pin hole perpendicular and center to the image sensor. The pin hole length for each imaging device in the 3-D imaging device system should be determined. The pin hole length (sometimes known as focal point length) is generally given in the manufacturer's specification of the imaging device. To obtain a more accurate pin hole length for the individual imaging devices, the following instrument may be used as illustrated in FIG. 9.

The accuracy of the pin hole length PL is important in that from the pin hole length, sampled points of a target object may be translated to a coordinate on the image sensor. For illustration purposes, the right imaging device 13 is used. The pin hole length PL of the imaging device 13 may be determined by placing the imaging device 13 on a rail 91 which has two rectangles 93, 95 that slide back and forth along the axis of the rail 91. Let the two rectangles 93, 95 be designated first rectangle 93 and second rectangle 95. Furthermore, at least one of the defining length of a rectangle (commonly referred to as horizontal or vertical length) should be known for each rectangle. In this example, the horizontal half length of the second rectangle 95 is known which is A and the horizontal half length of the first rectangle 93 is known which is B. The horizontal half length B should be made smaller than the horizontal half length A. The ratio R is then the ratio of horizontal half length B over horizontal half length A. Both rectangles should be mounted such that the center line of the rail 91 is normal to the two rectangles. Furthermore, the center of the rectangles 93, 95 should coincide with the center of the rail 91. The first rectangle 93 and the second rectangle 95 should further be parallel to each other. First rectangle 93 must furthermore be slidable and at all times comply with the requirements outlined above. By sliding the first rectangle 93 in the direction of the imaging device 13, while the second rectangle 95 remains stationary, at a certain distance on the rail 91, from the image sensor's 15 perspective, the projection of the rectangles' defining lengths will coincide on the image sensor. At this point, the image of horizontal length B of the first rectangle 93 and the image of horizontal length A of the second rectangle 95 passes through the pin hole P to project a same length X on the same location of the image sensor 15 as illustrated in FIG. 6. Knowing the distance between the first rectangle 93 and second rectangle 95 which is C measured on the rail 91, and the length of the projection on the image sensor measured by X, the pin hole length PL may be defined by the formula

$$PL = X \cdot C / (A \cdot (1 - R))$$

Generally, the measurement inside the imaging device is determined in pixel units. Note that pin hole length PL obtains its dimension from projection X. Since projection X is measured in pixel units, pin hole length PL is also measured in pixel units which is adequate in practice.

Referring back to FIG. 6, block 62 illustrates determining Cx and Cy of the image sensor. FIG. 8b illustrates an image of the calibration target 75 captured by an image sensor 15. From the captured image, image sensor 15 is computationally moved so that the center 72 of the image sensor coincides with the imaged center 77 of the calibration target 75. The movement in pixel units in the X'-axis and in the Y'-axis with respect to the imaging device coordinate corresponds to the Cx and Cy respectively. Alternatively, because the pin hole length is known as described previously, Cx and Cy may be also defined in terms of the angular rotation in the X'-axis and Y'-axis with respect to the pin hole P. When the target center 77 coincides with the center 72 of the image sensor 15, the center of the image is aligned with axis M (see FIG. 7) passing through the pin hole and the origin of the chosen coordinate system. Through Cx and Cy, the orientation of the image sensor 15 may be determined with respect to axis M and the pin hole P. As the calibration target 77 is relocated to the center 72 of the image sensor 15 from the image sensor's perspective, the corner dots 76 representing the calibration target are also computationally moved to represent the view from the image sensor in its new position. The relocated calibration target is illustrated by FIG. 8c.

Below is an exemplary fragmented program in Visual Basic that illustrates the above description.

For the example, the following definitions are required:

VERTEX2D is describing one point of an image. Its relevant members are the x and y coordinate (i.e. dx, dy);

VTXLIST2D is a list of VERTEX2D objects with associated housekeeping storage (i.e. calibration target where member 0 is center point, member 1 is top point, member 2 is bottom point, member 3 is left point, member 4 is right point);

VERTEX3D is describing one point in 3D space. Its relevant members are the x, y, and z coordinates (dx, dy, dz); and

VTXLIST3D is a list of VERTEX3D objects.

```
Public Sub gs_UndoCoxCoyWjm(InVec As VTXLIST2D, res As VTXLIST2D, Coff As VERTEX2D)
'the offsets are returned in Coff
Dim v1 As VERTEX3D
Dim v2 As VERTEX3D
Dim v3 As VERTEX3D
Dim vtx2tmp1 As VERTEX2D
Dim vRot As VERTEX3D
Dim dCosAngle As Double
Dim dAngle As Double
Dim i As Integer
'figure out the x and y offsets
Coff.dx=v2.dx
Coff.dy=v2.dy
'work to simulate the effect of un-doing the off-center of the
   imaging device
'get the normal and rotation angle
v1.dx=0
v1.dy=0
v1.dz=f2_dCAMF
v2.dx=InVec.vx2A(0).dx
v2.dy=InVec.vx2A(0).dy
v2.dz=f2_dCAMF
Call gs_CosAngle3(v1, v2, dCosAngle) 'get the cosine of
   angle between v1 and v2
Call gs_ArcCos(dCosAngle, dAngle) 'get the angle from
   the cosine
Call gs_orth3(v1, v2, vRot) 'get an orthogonal vector to the
   plane spanned by v1 and v2. That is the vector around
```

which the picture has to be rotated to bring the center point into the center of the image plane.

```
'rotate all vectors in the target vertices list to undo the Cx
    and Cy effect
For i=0 To InVec.1NumVertex-1
v1.dx=InVec.vx2A(i).dx
v1.dy=InVec.vx2A(i).dy
v1.dz=dCAMF 'dCAMF is the distance from the image
    sensor to the pin hole locations (in pixels)
Call gs_rot3dVec(dAngle, vRot, v1, v2)
Call gs_project3Planar(f2_dCAMF, v2, v3)
res.vx2A(i).dx=v3.dx
res.vx2A(i).dy=v3.dy
res.vx2A(i).bFlag=True
Next i
End Sub
```

Block 63 of FIG. 6 illustrates determining the roll of the image sensor. The roll is the rotation of the image sensor around the Z'-axis of the imaging device coordinate system, in this instance, the Z'-axis corresponding to the axis M passing through the pin hole P and the center 72 of the image sensor 15. Referring to FIG. 8*c*, once the image sensor is relocated so that the image sensor center 72 coincides with target center 77, the projection of the Y-axis of the target 75 onto the image sensor is compared with a Y'-axis passing through the center and parallel to vertical sides of the image sensor 15 from the image sensor's perspective. The angle deviation between the Y-axis and the Y'-axis is the roll of the image sensor. The roll is compensated for by computationally rotating the image sensor along the axis M until the Y-axis of the calibration target is parallel with the Y'-axis of the image sensor. As the roll is compensated, the corner dots representing the calibration target are also computationally moved to represent the view from the image sensor's perspective due to the roll compensation. The roll compensated calibration target 75 is illustrated by FIG. 8*d*.

Below is an exemplary program in Visual Basic that illustrates the description above:

```
Public Sub gs_UndoRollWjm(src As VTXLIST2D, res As
    VTXLIST2D, dTwist As Double) 'dTwist is the detected
    roll angle'
'undo the roll after the imaging device orientation has been
    corrected for Cx,Cy
Dim dalpha1 As Double
Dim dalpha2 As Double
Dim dAlpha As Double
Dim v1 As VERTEX3D
Dim v2 As VERTEX3D
Dim i As Integer
'rotation around the z axis with angle defined by atn x/y
dalpha1=Atn(src.vx2A(1).dx/src.vx2A(1).dy)
dalpha2=Atn(src.vx2A(2).dx/src.vx2A(2).dy)
dAlpha=(dalpha1+dalpha2)/2 'take arithmetic mean
dTwist=-dAlpha*190/const_PI
'simulate undoing the roll on the five calibration points of
    the image
For i=LBound(src.vx2A) To UBound(src.vx2A)
v1.dx=src.vx2A(i).dx
v1.dy=src.vx2A(i).dy
v1.dz=0
Call gs_rot3dZ(dAlpha, v1, v2)
res.vx2A(i).dx=v2.dx
res.vx2A(i).dy=v2.dy
Next i
End Sub
```

Block 64 of FIG. 6 illustrates determining the latitude position of the pin hole. Latitude determines the vertical position of the pin hole with respect to the calibration target. The latitude of the pin hole is determined by the length between the top and center dot and the length between the bottom and the center dot of the calibration target. The image sensor is computationally moved radially around the center point of the calibration target in a plane defined by the Y-axis and the pin hole location. The orientation of the imaging device maintains the condition that the center of the image sensor remains on the axis defined by the center of the calibration target and the relocating pin hole. The image sensor is moved until the length between the top and center dot and the length between the bottom and center dot are equal as viewed by the image sensor. At this point, the pin hole has computationally been moved into the X-Z plane and the angle the image sensor has computationally moved is the latitude. The corner dots representing the calibration target are computationally transformed to represent the view from the image sensor's perspective due to the positional change of the image sensor.

Below is an exemplary program in Visual Basic that illustrates the description above:

```
Public Sub gs_UndoLatitudeWjm(src As VTXLIST2D, res
    As VTXLIST3D, proj As VTXLIST2D, dLatitude As
    Double)
'Find out the latitude through comparison of the angles from
    midpoint in both
'directions of y
Dim dm1 As Double
Dim dm2 As Double
Dim dm3 As Double
Dim v1 As VERTEX3D
Dim v2 As VERTEX3D
Dim v3 As VERTEX3D
Dim v4 As VERTEX3D
Dim i As Integer
Dim dAlpha As Double
dm1=src.vx2A(1).dy/f2_dCAMF
dm2=src.vx2A(2).dy/f2_dCAMF
If Abs(dm1+dm2)>0.000000000001 Then
dm3=2*dm1*dm2/(dm1+dm2)
dAlpha=Atn(dm3)-const_PI/2
Else
dm3=1E+100
dAlpha=0
End If
'range of dalpha is -90 to +90 deg
If dAlpha<-const_PI/2 Then
dAlpha=dAlpha+const_PI
End If
dLatitude=dAlpha*190/const_PI
Dim vpLatVec As VERTEX3D
Dim vp1 As VERTEX3D
Dim vp2 As VERTEX3D
Dim vp3 As VERTEX3D
Dim vp4 As VERTEX3D
Dim vp5 As VERTEX3D
Dim vl1 As VERTEX3D
Dim vl2 As VERTEX3D
Dim vPt As VERTEX3D
'correct the display:
'create a vector which is tilted into the direction of the
    latitude
vpLatVec.dx=0
vpLatVec.dy=Cos(dAlpha)
vpLatVec.dz=Sin(dAlpha)
vp1.dx=0
vp1.dy=0
```

```
vp1.dz=0
Call gs_vec3ToNormalPlane(vpLatVec, vp2, vp3)
vp4.dx=1
vp4.dy=0
vp4.dz=0
vp5.dx=vpLatVec.dx
vp5.dy=vpLatVec.dy
vp5.dz=vpLatVec.dz
'shift the plane from the pin hole to the center of the ccd
vp1.dz=vp1.dz-f2_dCAMF
vp2.dz=vp2.dz-f2_dCAMF
vp3.dz=vp3.dz-f2_dCAMF
vp4.dz=vp4.dz-f2_dCAMF
vp5.dz=vp5.dz-f2_dCAMF
vl1.dx=0
vl1.dy=0
vl1.dz=0
res.vx3A(0).dx=src.vx2A(0).dx
res.vx3A(0).dy=src.vx2A(0).dy
res.vx3A(0).dz=-f2_dCAMF
'simulate un-doing the latitude
For i=1 To 4
vl2.dx=src.vx2A(i).dx
vl2.dy=src.vx2A(i).dy
vl2.dz=-f2_dCAMF
If i<3 Then
Call gf_PlaneLineIntersection(vp1, vp4, vp5, vl1, vl2, vPt)
Else
Call gf_bPlaneLineIntersection(vp1, vp2, vp3, vl1, vl2, vPt)
End If
'rotate around the x axis
vPt.dz=vPt.dz+f2_dCAMF
Call gs_rot3dX(-dAlpha, vPt, v3)
'shift everything back by the f distance
v3.dz=v3.dz-f2_dCAMF
res.vx3A(i)=v3
'project into the image sensor plane
Call gs_project3Planar(-f2_dCAMF, v3, v4)
proj.vx2A(i).dx=v4.dx
proj.vx2A(i).dy=v4.dy
Next i
End Sub
```

Block 65 of FIG. 6 illustrates determining longitude position of the pin hole. Longitude determines the position of the pin hole in the X-axis with respect to the chosen coordinate system. Longitude is determined by the length between the left dot and the center dot and the length between the right dot and the center dot of the calibration target as imaged in the image sensor. The image sensor is computationally moved radially around the calibration target in the X-Z plane. The orientation of the imaging device is changed so that the center of the image sensor remains on the axis defined by the center of the calibration target and the relocating pin hole. The image sensor is moved until the length between the left dot and the center dot is equal to the length between the right dot and the center dot as viewed by the image sensor. At this point, the angle the pin hole has moved is the longitude. The corner dots representing the calibration target are computationally transformed to represent the view from the image sensor's perspective due to the positional change of the image sensor.

Below is an exemplary program in Visual Basic that illustrates the description above:

```
Public Sub gs_UndoLongitudeWjm(src As VTXLIST3D, res As VTXLIST3D, proj As VTXLIST2D, dLongitude As Double)
'Find out the longitude through comparison of the angles from midpoint in both
'directions of x
Dim dm1 As Double
Dim dm2 As Double
Dim dm3 As Double
Dim v1 As VERTEX3D
Dim v2 As VERTEX3D
Dim v3 As VERTEX3D
Dim v4 As VERTEX3D
Dim i As Integer
Dim dA As Double
Dim dAlpha As Double
'first get the projection of point 3 into the yz plane
Call gs_project3Planar(-f2_dCAMF, src.vx3A(3), v1)
Call gs_project3Planar(-f2_dCAMF, src.vx3A(4), v2)
'next find out what angle we have from 0 point to point 3
'in the xz plane
dm1=v1.dx/v1.dz
dm2=v2.dx/v2.dz
If Abs(dm1+dm2)>0.000001 Then
dm3=2*dm1*dm2/(dm1+dm2)
Else
dm3=1000000
End If
dAlpha=const_PI/2-Atn(dm3)
If dAlpha>const_PI/2 Then dAlpha=dAlpha-const_PI/2
dLongitude=dAlpha*190/const_PI
'simulate undoing of longitude
For i=0 To 4
v2=src.vx3A(i)
'simulate shift into the pin hole plane, by making z 0
v2.dz=v2.dz+f2_dCAMF
'rotate around the y axis
Call gs_rot3dY(dAlpha, v2, v3)
'shift everything back by the f distance
v3.dz=v3.dz-f2_dCAMF
res.vx3A(i)=v3
'project into the image sensor plane
Call gs_project3Planar(-f2_dCAMF, v3, v4)
proj.vx2A(i).dx=v4.dx
proj.vx2A(i).dy=v4.dy
Next i
End Sub
```

Block 66 of FIG. 6 illustrates determining the radius of the pin hole. Radius is the distance between the pin hole and the origin of the chosen coordinate system. Radius may be determined, for example, in the following manner. From the calibration target, the distance between the corner dot and the center dot of the calibration target is known. From the image sensor, the imaged corner dot and the center dot of the calibration target may be measured in pixels. Because the pin hole length is known, the radius may be defined by the formula

Radius=PL*(A'/X')

where A' is the distance between a corner dot and a center dot of the calibration target and X' is the imaged distance of the corner dot and the center dot at the image sensor.

Below is an exemplary program in Visual Basic that illustrates the description above:

```
Public Sub gs_DetermineDistanceWjm(src As VTXLIST3D, res As VTXLIST3D, proj As VTXLIST2D, dDistance As Double)
res=src
End Sub
```

In the instances where the change in the lengths corresponding to the relative dots in the calibration target 75 are

relatively minor, it is desirable to install an additional calibration dot that is protruding from the plane where the calibration target 75 is located to be closer to the image sensor with respect to the remaining target dots 76, 77. Because the additional dot is closer to the image sensor, the dot is more susceptible to the change in the direction of the image sensor than the dots 76, 77 on the calibration target 75. FIG. 10 illustrates an exemplary calibration target 101 having a diamond shape with a dot 108 protruding normally from the plane of the calibration target 101. This is achieved by having a triangular plane protruding from the plane where the calibration target 101 is located as illustrated. The angle of the protrusion may be 45°. A calibration dot 108 is placed on the protruding triangle in a manner such that the dot 108 aligns with the left dot 102 and right dot 104 of the calibration target 101.

Referring back to the determination of the latitude of block 64 in FIG. 6, the image sensor having a latitude with respect to the calibration target 101 will detect the protruding dot 108 to be either above or below an X-axis passing through the left dot 102 and right dot 104 of the calibration target 101. The image sensor is computationally moved vertically until the dot 108 aligns with the left dot 102 and the right dot 104. At this point, the angular distance the pin hole has moved corresponds to the latitude. Turning to the determination of longitude of block 65 in FIG. 6, the image sensor having a longitude with respect to the calibration target 101 will detect the protruding dot 108 to be at a distance from the calibration target 101. As an example, the ratio corresponding to the distance between the protruding dot 108 and the right dot 104 in the plane where the calibration target 101 is located and the distance between the right dot 104 and the left dot 102 can be predetermined, thus, the ratio is computationally calculated as the image sensor is computationally moved horizontally until the predetermined ratio is reached. At this point, the angular distance the pin hole has moved corresponds to the longitude.

When the six positional values are known for each image sensor of the imaging devices in the 3-D imaging device system, the calibration is complete. The computing device used to create 3-D data is able to know the relative position of the image sensors in the chosen coordinate system by reverse sequencing the positional values obtained. For example, if the sequence taken is Cx, Cy, roll, latitude, longitude, radius, then the computing device by reverse sequencing, that is computing in the sequence, radius, longitude, latitude, roll, Cy, Cx, knows the position of the image sensor.

With the calibration performed on each of the imaging devices in the 3-D imaging device system and before stereoscopic images are taken by the imaging devices 12 and 13 as illustrated in FIG. 1, for example, initially, markers should be placed on the target object 20 that outline the contours or features of the object as illustrated in FIG. 3*a*. For example, vertical lines 21 may be evenly placed on object 20. However, it should be noted that more lines 21 may be added to object 20 in areas where there are fine features such as the eyes or the nose, for example. The vertical lines 21 may be painted onto the object 20 using fluorescent paint that is visible only in the dark such that structural data may be obtained in the dark. Alternatively, the paint used may be invisible in the visible spectrum but visible to radiation outside the visible spectrum such as infra-red or ultraviolet light. Alternatively, the vertical lines 21 may be projected onto the object 20 using light source 16 such as infra-red laser or visible laser. It will be appreciated that the description of vertical lines should by no means be interpreted as limiting the scope of the invention. For example, horizontal lines may be used depending on the orientation of the imaging devices 12 and 13. Furthermore, depending on the sophistication of the diffracting device 17, a grid of uniform dots may be projected onto the object.

Using structured light as an example to further the understanding of the invention, the light source 16 with the diffracting device 17 project a structured light, in this example, vertical lines onto the target object. With the structured light projected, a stereoscopic image of the object is captured by the left imaging device 12 and right imaging device 13 of the 3-D imaging device system 10. FIG. 11 is a flowchart of an embodiment illustrating the extraction of 3-D data from a stereoscopic image of an object comprising a left image and a right image. Block 112 illustrates determining the number of vertical lines detected by the right image sensor corresponding to the vertical lines projected onto the target object using structured light. Block 114 illustrates determining the number of vertical lines detected by the left image sensor also corresponding to the vertical lines projected onto the target object. Block 116 illustrates matching the vertical lines detected by the right image sensor with the vertical lines detected by the left image sensor in a correlating manner. Block 118 illustrates setting a first counter at 1 reflecting the first correlating vertical line detected at the right image sensor. Block 120 illustrates converting the correlating vertical line into a plurality of dots. Typically, the distance between the dots is measured in pixel units, for example, a dot from the vertical line may be formed per every five pixels. Note that by controlling the number of correlating vertical lines and the number of dots to be converted from each vertical line, a desired number of points in the 3-D data may be obtained. Block 122 illustrates determining the number of dots converted from the vertical line and setting a second counter at 1 reflecting the designated dot to be the first dot. Block 124 illustrates computing a "line of sight" originating from the designated dot and passing through the pin hole of the right imaging device to project the line in 3-D space. At a certain point in the line of sight, the coordinate unknown, the point will intersect with a vertical line in the target object that correlates with the vertical line from which the designated dot originated. The intersecting point at the target object will also correspond to the position of the designated dot in the right image sensor. In one embodiment, the line of sight for the right image sensor may be produced in the following manner.

Referring to FIG. 7, the position of the pin hole P of the right imaging device 15 is known in the chosen coordinate system from the calibration of the right imaging device. Note that a pin hole P is positioned at a fixed distance PL from the center and normal to the image sensor. In this example, let the image sensor 15 comprise a 640x480 pixel array. However, one skilled in the art will recognize that other array sizes may also be used. Using pin hole P as the origin of an imaging device coordinate system, every point in the image sensor 15 may be referenced from the pin hole P in pixel values ($\Delta X'$, $\Delta Y'$, PL) where $\Delta X'$ is the deviation in the X'-axis of the position of the designated dot from the center of the image sensor 15, $\Delta Y'$ is the deviation in the Y'-axis of the position of the designated dot from the center of the image sensor 15, PL is a known fixed value which is a fixed distance in the Z'-axis, all three values measured in pixel units. Because the pin hole coordinate is known in the chosen coordinate system using this coordinate and values ($\Delta X'$, $\Delta Y'$, PL) corresponding to the designated dot, an equation corresponding to the line of sight for the designated dot may be produced. FIG. 12 illustrates a line of sight L1 representing the designated dot of the right image sensor 15.

Block **126** illustrates determining a point in the correlating vertical line of the left image sensor **14** that corresponds to the designated dot of the right image sensor **15**. Once the corresponding point is determined, because the coordinate of the left pin hole is known in the chosen coordinate system through calibration, and the positional value of the point may be determined in the manner described immediately above, an equation representing the light of sight of the corresponding point may be obtained. FIG. **12** illustrates a line of sight L2 representing the corresponding point of the left image sensor **14**. In one embodiment, the corresponding point in the left image **14** may be determined by "tracing" the line of sight L1 of the right image sensor **15** onto the left image sensor **14**. Because the correlating vertical lines in the left and right image sensor corresponds to the same vertical line in the target object, the point of intersection of the line of sight from the right image sensor and the vertical line in the target object as viewed by the left image sensor would be the corresponding point of the designated dot to the right image sensor. An analogy will be used to explain block **126** to aid in the understanding of the procedure.

Assuming that the designated dot D1 in the vertical line imaged at the right image sensor **15** is able to emit a beam of light L1 that passes through right pin hole PR of the imaging device **13** and through space. The left imaging device **12** would be able to detect this beam of light L1 on its image sensor **14** from its field of view. The intersection of the detected beam of light L1 and the vertical line in the left image sensor **14** correlating with the vertical line in the right image sensor **15** from which the designated dot originated may be determined. The intersection point D2 would be the corresponding point of the designated dot D1 in the left image sensor. From the intersection point D2, a beam of light L2 may be projected that passes through the left pin hole PL of the left imaging sensor which should intersect with the beam of light L1. The point of intersection of the two beams of light L1 and L2 will be the X, Y, Z, coordinate in the chosen coordinate system of a point in the target object corresponding to the designated dot in the right image sensor which is one point of 3-D data of the target object. Of course, in reality, the above description is performed mathematically.

Below is an exemplary program written in Visual Basic to be used to illustrate a mathematical procedure performing the above description.

```
tc_MakeLineOfSight vx2A, ln3A, objA
'The first procedure above illustrates producing a right line of sight from the right image sensor where the coordinates of the right line of sight corresponds to the chosen coordinate system from calibration'
tc_Ln3ObjMakeLn2 ln3A, ln2A, objB
'The second procedure above illustrates converting the right line of sight from the right image sensor in the chosen coordinate system to a left imaging device coordinate system. The right line of sight is traced onto the left image sensor'
If gf_bContourLineIntersection(ln2A, cn2B, vx2B)
'The third procedure above illustrates finding the intersection of the traced right line of sight on the left image sensor and the vertical line imaged at the left image sensor correlating with the vertical line on the right image sensor from which the right line of sight originated'
Call tc_MakeLineOfSight(Vx2B, ln3B, objB)
'The fourth procedure above illustrates producing a left line of sight from the left image sensor from the intersecting point and the coordinate of the left pin hole of the left imaging device in the chosen coordinate system'
If gf_bLineDistance3D(ln3A, ln3B, vx3A, dDist)=False Then 'result in vx3A
'The fifth procedure above illustrates finding the intersection of the right and left line of sights, wherein if the intersection does not occur, the point where the right line of sight is closest to the left line of sight is designated as the intersection point'
```

The subroutine below illustrates the first procedure in further details:

```
Public Sub tc_MakeLineOfSight(vx2In As VERTEX2D, ln3Out As LINE3D, obj2D As Object)
Dim vx2A As VERTEX2D
Dim dDist As Double
dDist=50
vx2A.dx=vx2In.dx-obj2D.f2_iCAMCx/2 'obj2D 'determines the center X'-axis of the image sensor'
vx2A.dy=vx2In.dy-obj2D.f2_iCAMCy/2 'determines the center Y'-axis of the image sensor'
'Above two lines illustrate determining the center of the right image sensor and designating as coordinate (0, 0) in the right imaging device coordinate system'
ln3Out.vx3A.dx=obj2D.f2_dLOSStart*vx2A.dx/obj2D.f2_dCAMF
ln3Out.vx3A.dy=obj2D.f2_dLOSStart*vx2A.dy/obj2D.f2_dCAMF
ln3Out.vx3A.dZ=-obj2D.f2_dLOSStart
'Above three lines illustrate designating a starting point for the right line of sight, the coordinate defined in the right imaging device coordinate system'
ln3Out.vx3B.dx=obj2D.f2_dLOSEnd*vx2A.dx/obj2D.f2_dCAMF
ln3Out.vx3B.dy=obj2D.f2_dLOSEnd*vx2A.dy/obj2D.f2_dCAMF
ln3Out.vx3B.dZ=-obj2D.f2_dLOSEnd
'Above three lines illustrate designating an ending point for the right line of sight, the coordinate defined in the right imaging device coordinate system'
'Note the starting point and the ending point may be user defined so that the distance between the starting point and the ending point is sufficient to intersect the target object. As an example if the target object is 4 feet from the right image sensor, the starting point may be designated at 0.5 feet and the ending point may be designated at 6 feet'
Call gs_XformCameraToWorld(ln3Out.vx3A, ln3Out.vx3A, obj2D)
Call gs XformCameraToWorld(ln3Out.vx3B, ln3Out.vx3B, obj2D)
'Transforming the coordinates of the starting point and the ending point from coordinates in the right imaging device coordinate system to coordinates in the chosen coordinate system'
End Sub
'The subroutine below illustrates converting points in the imaging device coordinate system to coordinates in the chosen coordinate system'
Public Sub gs_XformCameraToWorldWjm(vx3In As VERTEX3D, vx3Out As VERTEX3D, obj2D As Object)
Dim v1 As VERTEX3D
Dim v2 As VERTEX3D
Dim vRot As VERTEX3D
Dim dCosAngle As Double
Dim dAngle As Double
Dim dTwist As Double
Dim dLongitude As Double
Dim dLatitude As Double
Dim dDistance As Double
Call gs_rot3dCOxCOyWjm(vx3In, obj2D, False, v2)
```

v1=v2

'The call routine above compensates for the orientation Cx and Cy values of the image sensor to transform the right imaging device coordinate system to the chosen coordinate system. The compensation of the Cy value which may be obtained by an angular rotation in the X'-axis which changes the position of the Y'-Z' plane. The starting point and the ending point of the right line of sight is compensated by the angular rotation amount for the change in the Y'-Z' plane. The compensation of the Cx value may be obtained by the angular rotation in the Y'-axis which changes the position of the X'-Z' plane. The starting point and the ending point of the right line of sight is compensated by the angular rotation amount for the change in the Y'-Z' plane. When the orientation Cx and Cy is compensated for the right imaging device, the center of the right imaging sensor is aligned with the pin hole of the right imaging device and the origin of the chosen coordinate system. The new position of the starting point and the ending point of the right line of sight reflects the perception of the points from the changed position of the right image sensor.'

dTwist=obj2D.f2_dCAMTtwist*const_PI/190

Call gs_rot3dZ(-dTwist, v1, v2)

v1=v2

'The call routine above compensates for the roll of the right image sensor with respect of the chosen coordinate system. The compensation of the roll value which may be obtained by an angular rotation in the Z'-axis which changes the position of the X'-Y' plane. The starting point and the ending point of the right line of sight is compensated by the angular rotation amount for the change in the X'-Y' plane. The new position of the starting point and the ending point of the right line of sight reflects the perception of the points from the changed position of the right image sensor.'

'move by f along z axis to move us into the center of the world

dDistance=obj2D.f2_dCAMZ

v1.dz=v1.dZ+dDistance

'Above three lines illustrate compensating for the radius of the transformed right imaging device coordinate system. Note that once Cx, Cy and roll has been compensated, the right imaging device coordinate system is aligned with the chosen coordinate system having the origin at the pin hole of the right imaging device. By moving the origin at the pin hole to the origin of the chosen coordinate system, the position of the starting point and the ending point of the right line of sight reflects the perception of the points with the right image sensor at the origin. This is performed by compensating for the radius, latitude angle and longitude angle of the right imaging device coordinate system to the coordinate system into the chosen coordinate system.'

dLatitude=obj2D.f2_dCAMTlat*const_PI/190

Call gs_rot3dX(-dLatitude, v1, v2)

v1=v2

'Above three lines illustrate compensating for the latitude angle.'

dLongitude=obj2D.f2_dCAMTlong*const_PI/190

Call gs_rot3dY(dLongitude, v1, v2)

v1=v2

'Above three lines illustrate compensating for the longitude angle.'

vx3Out=v2

End Sub

Below are exemplary codes for the subroutine call to compensate for the orientation Cx and Cy values of the image sensor to transform the right imaging device coordinate system to the chosen coordinate system.'

Public Sub gs_rot3dCOxCOyWjm(vx3In As VERTEX3D, obj2D As Object, forward As Boolean, vx3Out As VERTEX3D)

Dim vx2v1 As VERTEX2D

Dim vx2v2 As VERTEX2D

Dim vx3v1 As VERTEX3D

Dim vx3v2 As VERTEX3D

Dim vx3Rot As VERTEX3D

Dim dCosAngle As Double

Dim dAngle As Double

'create the corrected 2d coordinates

vx2v1.dx=obj2D.f2_iCAMCOx-obj2D.f2_iCAMCx/2

vx2v1.dy=obj2D.f2_iCAMCOy-obj2D.f2_iCAMCy/2

'undo cox,coy through a rotation around the normal which is spanned by pinhole,cox,coy, and pinhole,0,0 ((z,x,y) as unit vectors)'

'build 3d vectors for the two known points

vx3v1.dx=0

vx3v1.dy=0

vx3v1.dZ=obj2D.f2_dCAMF

vx3v2.dx=vx2v2.dx

vx3v2.dy=vx2v2.dy

vx3v2.dZ=obj2D.f2_dCAMF

'get the rotation angle and the normal vector

Call gs_CosAngle3(vx3v1, vx3v2, dCosAngle)

Call gs_ArcCos(dCosAngle, dAngle)

If bForward=False Then

dAngle=-dAngle

End If

Call gs_orth3(vx3v1, vx3v2, vx3Rot)

Call gs_rot3dVec(dAngle, vx3Rot, vx3In, vx3Out)

End Sub

'The subroutine below illustrates the second procedure in further details.'Public Sub tc_Ln3ObjMakeLn2(ln3A As LINE3D, ln2A As LINE2D, obj2D As Object)

Dim vx2A As VERTEX2D

Dim vx2B As VERTEX2D

Dim vx3AWorld As VERTEX3D

Dim vx3ACam As VERTEX3D

Dim vx3BWorld As VERTEX3D

Dim vx3BCam As VERTEX3D

'transform the 3D line into camera coordinates

vx3AWorld=ln3A.vx3A

vx3BWorld=ln3A.vx3B

Call gs_XformWorldToCamera(vx3AWorld, vx3ACam, vx2A, obj2D)

The call routine above transforms the starting point of the right line of sight in the chosen coordinate system into a coordinate in the left image sensor plane in the left imaging device coordinate system.'

Call gs_XformWorldToCamera(vx3BWorld, vx3BCam, vx2B, obj2D)

'The call routine above transforms the ending point of the right line of sight in the chosen coordinate system into a coordinate in the left image sensor plane in the left imaging device coordinate system.'

ln2A.vx2A=vx2A

ln2A.vx2B=vx2B

End Sub

'Below is an exemplary call routine to transform a point in the chosen coordinate system to a point in the imaging device coordinate system. The routine below may be applied to the starting point and the ending point of the right line of sight.'

Public Sub gs_XformWorldToCameraWjm(vx3In As VERTEX3D, vx3Out As VERTEX3D, obj2D As Object)

Dim dAlpha As Double

```
Dim dBeta As Double
Dim dF As Double
Dim dDistance As Double
Dim dLongitude As Double
Dim dLatitude As Double
Dim dTwist As Double
Dim vx3Rot As VERTEX3D
Dim iC0x As Integer
Dim iC0y As Integer
Dim iCx As Integer
Dim iCy As Integer
Dim vx3v1 As VERTEX3D
Dim vx3v2 As VERTEX3D
dLongitude=obj2D.f2_dCAMTlong*3.1415926/180
dLatitude=obj2D.f2_dCAMTlat*3.1415926/180
dTwist=obj2D.f2_dCAMTtwist*3.1415926/180
dF=obj2D.f2_dCAMF
dDistance=obj2D.f2_dCAMZ
Call gs_rot3dY(-dLongitude, vx3In, vx3v2)
The above call routine applies a longitude angle of the pin
  hole of the left imaging device to a point in the chosen
  coordinate system.'
vx3v1=vx3v2
Call gs_rot3dX(dLatitude, vx3v1, vx3v2)
'The above call routine applies a latitude angle of the pin
  hole of the left imaging device to the point in the chosen
  coordinate system.'
vx3v1=vx3v2
vx3v1.dZ=vx3v1.dZ-dDistance
'The above call routine applies a radius of the pin hole of the
  left imaging device to the point in the chosen coordinate
  system.'
Call gs_rot3dZ(dTwist, vx3v1, vx3v2)
'The above call routine applies a roll of the left image sensor
  to the point in the chosen coordinate system.'
vx3v1=vx3v2
'apply c0x,c0y
Call gs_rot3dC0xC0yWjm(vx3v1, obj2D, True, vx3v2)
'The above call routine applies a Cx and Cy of the image
  sensor to the point in the chosen coordinate system.'
vx3v1=vx3v2
vx3Out=vx3v2
End Sub
```

Note that once the coordinates of the starting point and the ending point in the right imaging device coordinate system have been transformed to coordinates in the left imaging device coordinate system, the pin hole of the left imaging device, which may be the origin of the left imaging device coordinate system may be used to project the transformed starting point and the ending point onto the left image sensor. Stated differently, from the known coordinates of the starting point and the ending point in the left imaging device coordinate system, the known coordinate of the left pin hole, two equations may be determined that passes through the pin hole and image sensor, the two equations corresponding to line of sight of the starting point and the ending point. From the known pin hole length of the left imaging device, the two points on the location of the left image sensor may be determined. From the two located points, a line may be traced on the left image sensor corresponding to the line of sight of the first imaging device perceived by the left image sensor.

At block **126**, the computing device determines the X, Y, Z coordinates of the corresponding point D2 (see FIG. 12) at the left image sensor **14** and using the known X, Y, Z coordinate of the left pin hole PL determined through calibration, an equation representing a line L2 passing through the corresponding point D2 and the left pin hole PL is derived. Block **128** illustrates the computing device calculating the X, Y, Z coordinate of a point on the object corresponding to the designated point at the right image sensor using the equations derived from block **124** and block **126**. Note that the equations have variables in terms of the pin hole coordinate which is defined in terms of a coordinate in the chosen coordinate system. Thus, the X, Y, Z coordinate of the point on the object would be a coordinate in the chosen coordinate system. Block **130** illustrates determining if there are any more points on the vertical line to be processed and if so, the counter is incremented and the steps illustrated in blocks **120–130** executed. Block **132** illustrates determining if there are any more vertical lines to be processed. Blocks **120–132** illustrates if there are more vertical lines to be processed, the counter in incremented and the process is repeated. Otherwise the extraction of the 3-D data is complete and the process halts.

The exemplary embodiments described herein are provided merely to illustrate the principles of the invention and should not be construed as limiting the scope of the invention. Rather, the principles of the invention may be applied to a wide range of systems to achieve the advantages described herein and to achieve other advantages or to satisfy other objectives as well.

What is claimed is:

1. A method of extracting three dimensional (3-D) data from a target object comprising:

performing calibration on a first imaging device and a second imaging device;

compensating a starting point and an ending point coordinate;

forming a plurality of markers on said target object;

capturing a plurality of images of said target object;

designating a first point from a first marker in a first image;

determining a first line equation corresponding to said first point;

determining a second point corresponding to said first point in a second marker in a second image;

determining a second line equation corresponding to said second point; determining an intersection of said first line equation and said second line equation.

2. The method according claim **1**, wherein forming a plurality of markers further comprises projecting a structured light onto said target object.

3. The method according to claim **2**, where said structured light is one of lines, grids, dots, segmented lines produced by a light source.

4. The method according to claim **3** wherein said light source is a laser emitter.

5. The method according to claim **4** further comprising coding said plurality of markers such that a marker and a successive marker have a different coding.

6. The method according to claim **1** wherein capturing a plurality of images of said target object further comprises:

capturing a first image having a first set of markers from said plurality of markers;

capturing a second image having a second set of markers from said plurality of markers where said second set of markers having at least one marker correlating with a marker in said first set of markers;

determining a number of said correlating markers.

7. The method according to claim **6**, wherein said plurality of markers are a plurality of lines, further comprising:

forming a plurality of points from said plurality of lines on one of said first and second images.

8. The method according to claim 1, wherein determining said first line equation further comprises:

determining a position of a first pin hole corresponding to said first imaging device capturing said first image;

determining a relative position of said first point with respect to said first pin hole;

determining said first line equation originating from said first point and passing through said first pin hole.

9. The method according to claim 8, wherein said position of said first pin hole is respective to a first imaging device coordinate system, the method further comprising:

selecting a starting point of said first line equation having a starting point coordinate in said first imaging device coordinate system;

selecting an ending point of said first line equation having a ending point coordinate in said first imaging device coordinate system;

transforming said starting point coordinate from said first imaging device coordinate system to a starting point coordinate in a chosen coordinate system;

transforming said ending point coordinate from said first imaging device coordinate system to an ending point coordinate in said chosen coordinate system.

10. The method according to claim 9, wherein transforming said starting point coordinate and transforming said ending point coordinate in said first imaging device coordinate system to said chosen coordinate system comprises:

compensating said starting point coordinate and said ending point coordinate for an orientation Cx obtained from calibration of said first imaging device;

compensating said starting point coordinate and said ending point coordinate for an orientation Cy obtained from calibration of said first imaging device;

compensating said starting point coordinate and said ending point coordinate for a roll obtained from calibration of said first imaging device;

compensating said starting point coordinate and said ending point coordinate for a latitude angle obtained from calibration of said first imaging device;

compensating said starting point coordinate and said ending point coordinate for longitude angle obtained from calibration of said first imaging device;

compensating said starting point coordinate and said ending point coordinate for a radius obtained from calibration of said first imaging device.

11. The method according to claim 9, wherein determining a second point corresponding to said first point in said second marker in said second image further comprises:

transforming said starting point coordinate from said chosen coordinate system to a starting point coordinate in a second imaging device coordinate system;

transforming said ending point coordinate from said chosen coordinate system to an ending point coordinate in said second imaging device coordinate system;

projecting said transformed starting point in said second imaging device coordinate system on a second image sensor plane;

projecting said transformed ending point in said second imaging device coordinate system on said second image sensor plane;

determining an intersection point of a third line equation passing through said projected starting point and said projected ending point projected on said image sensor plane and said second marker correlating with said first marker in which said first point originated;

designating said intersection point as said second point.

12. The method according to claim 11, wherein transforming said starting point coordinate and transforming said ending point coordinate in said chosen coordinate system to said second imaging device coordinate system comprises:

applying to said starting point coordinate and said ending point coordinate a radius obtained during calibration of said second imaging device;

applying to said starting point coordinate and said ending point coordinate a longitude angle obtained from calibration of said second imaging device;

applying to said starting point coordinate and said ending point coordinate a latitude angle obtained from calibration of said second imaging device;

applying to said starting point coordinate and said ending point coordinate a roll obtained from calibration of said second imaging device;

applying to said starting point coordinate and said ending point coordinate an orientation Cy obtained from calibration of said second imaging device;

applying to said starting point coordinate and said ending point coordinate an orientation Cx obtained from calibration of said second imaging device.

13. The method according to claim 11, wherein projecting said starting point and projecting said ending point in said second imaging device coordinate system on a second image sensor plane further comprises:

determining a fourth line equation passing through said starting point coordinate in said second imaging device coordinate system and a second pin hole of said second imaging device;

determining a fifth line equation passing through said starting point coordinate in said second imaging device coordinate system and said second pin hole of said second imaging device;

determining projection of said transformed starting point in said second imaging device coordinate system on said second image sensor plane using said fourth line equation and a pin hole length of said second imaging device;

determining projection said transformed ending point in said second imaging device coordinate system on said second image sensor plane using said fifth line equation and said pin hole length of said second imaging device.

14. The method according to claim 1, wherein determining said second line equation further comprises:

determining a position of a second pin hole corresponding to said second imaging device capturing said second image;

determining a relative position of said second point with respect to said second pin hole;

determining said second line equation originating from said second point and passing through said second pin hole.

15. The method according to claim 1, wherein determining intersection of said first line equation and said second line equation further comprises:

determining if said intersection occurs between said first line equation and said second line equation;

if said intersection does not occur then determining a third point in said first line equation and a fourth point in said

second line equation where a distance between said first line equation and said second line equation is at a minimum compared with remaining points in said first line equation and said second line equation;

selecting one of said third point and fourth point as an intersection point.

16. The method of claim 1, wherein said calibration comprises:

determining a pin hole length of said first imaging device;

determining an orientation C_x of said first imaging device;

determining an orientation C_y of said first imaging device;

determining a roll of said first imaging device;

determining a latitude of a pin hole corresponding to said first imaging device;

determining a longitude of said pin hole corresponding to said first imaging device; and

determining a radius of said pin hole corresponding to said first imaging device.

17. The method of claim 16, further comprising:

determining a pin hole length of said second imaging device;

determining an orientation C_x of said second imaging device;

determining an orientation C_y of said second imaging device;

determining a roll of said second imaging device;

determining a latitude of a pin hole corresponding to said second imaging device;

determining a longitude of said pin hole corresponding to said second imaging device; and

determining a radius of said pin hole corresponding to said second imaging device.

18. The method of claim 17, wherein said pin hole length, said orientation C_X, said orientation C_Y, said roll, said latitude and said longitude are used for compensating said starting point and said ending point coordinate.

* * * * *

US006466701B1

(12) **United States Patent**
Ejiri et al.

(10) **Patent No.: US 6,466,701 B1**
(45) **Date of Patent: Oct. 15, 2002**

(54) **SYSTEM AND METHOD FOR DISPLAYING AN IMAGE INDICATING A POSITIONAL RELATION BETWEEN PARTIALLY OVERLAPPING IMAGES**

(75) Inventors: **Koichi Ejiri**, Chiba; **Shin Aoki**, Kanagawa; **Takashi Saitoh**, Kanagawa; **Guan Haike**, Kanagawa; **Takuji Sakamoto**, Tokyo, all of (JP)

(73) Assignee: **Ricoh Company, Ltd.**, Tokyo (JP)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/150,288**

(22) Filed: **Sep. 9, 1998**

(30) **Foreign Application Priority Data**

Sep. 10, 1997 (JP) 9-245522

(51) **Int. Cl.**[7] **G06K 9/32**; G06K 9/36; H04N 5/228; H04N 5/76; H04N 7/18

(52) **U.S. Cl.** **382/284**; 382/294; 382/293; 382/291; 382/287; 382/282; 348/222; 348/231; 348/64; 348/49

(58) **Field of Search** 382/284, 318, 382/131, 282, 287, 291, 293, 294; 345/115, 345; 348/584, 601, 588, 222, 231, 239, 49, 50, 53, 56

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,140,647 A * 8/1992 Ise et al. 382/41
5,659,823 A * 8/1997 Mukai et al. 396/291
5,682,198 A * 10/1997 Katyayama et al. 348/445
5,721,624 A * 2/1998 Kumashiro et al. 358/450
5,845,166 A * 12/1998 Fellegara et al. 396/429
5,963,664 A * 10/1999 Kumar et al. 382/154
5,978,016 A * 11/1999 Lourette et al. 438/64
5,982,941 A * 11/1999 Loveridge et al. 382/260
5,982,951 A * 11/1999 Katayama et al. 382/284

FOREIGN PATENT DOCUMENTS

JP	5-161050	6/1993
JP	8-4783	2/1996
JP	9-266561	10/1997

* cited by examiner

Primary Examiner—Leo Boudreau
Assistant Examiner—Yosef Kassa
(74) *Attorney, Agent, or Firm*—Dickstein Shapiro Morin & Oshinsky LLP

(57) **ABSTRACT**

A camera system includes a display monitor which displays an image of an object, taken by an optical unit, on a screen of the monitor. A reading unit reads a preceding image and a current image among a plurality of partially overlapping images, from a memory device, the preceding image and the current image containing a common element. A determining unit determines a positional relation between the preceding image and the current image based on a common pattern derived from the common element in the two adjacent images read by the reading unit. A displaying unit displays an image indicating a boundary of the preceding image on the screen of the monitor at a shifted position according to the positional relation determined by the determining unit, with the current image concurrently displayed on the screen of the monitor.

18 Claims, 8 Drawing Sheets

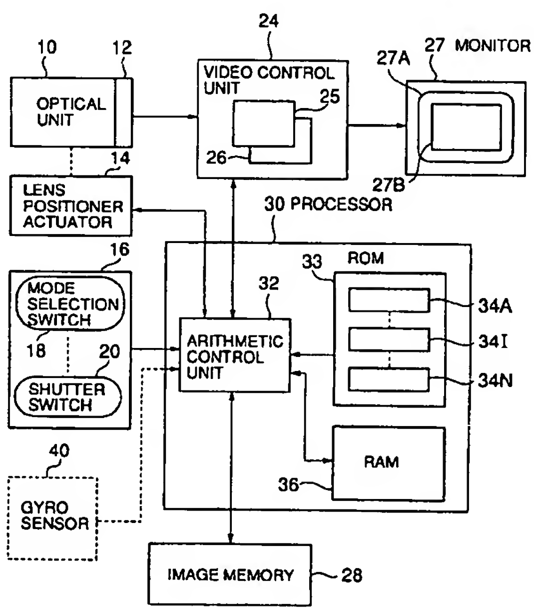


FIG.1

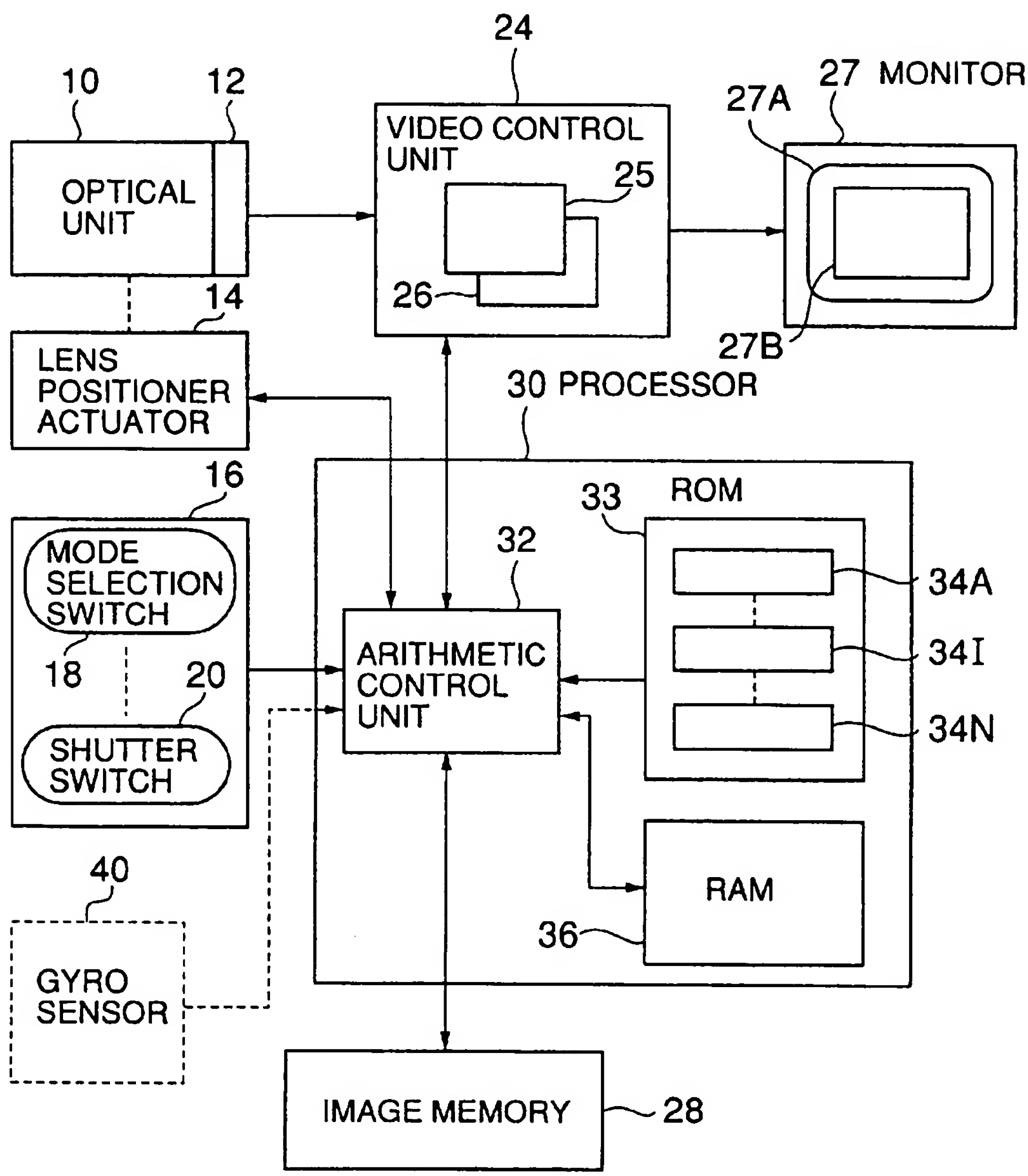

24
10
12
27 MONITOR
27A
VIDEO CONTROL UNIT
25
OPTICAL UNIT
26
14
27B
LENS POSITIONER ACTUATOR
30 PROCESSOR
16
33
ROM
MODE SELECTION SWITCH
32
34A
ARITHMETIC CONTROL UNIT
34I
18
20
34N
SHUTTER SWITCH
40
RAM
36
GYRO SENSOR
IMAGE MEMORY
28

FIG.2

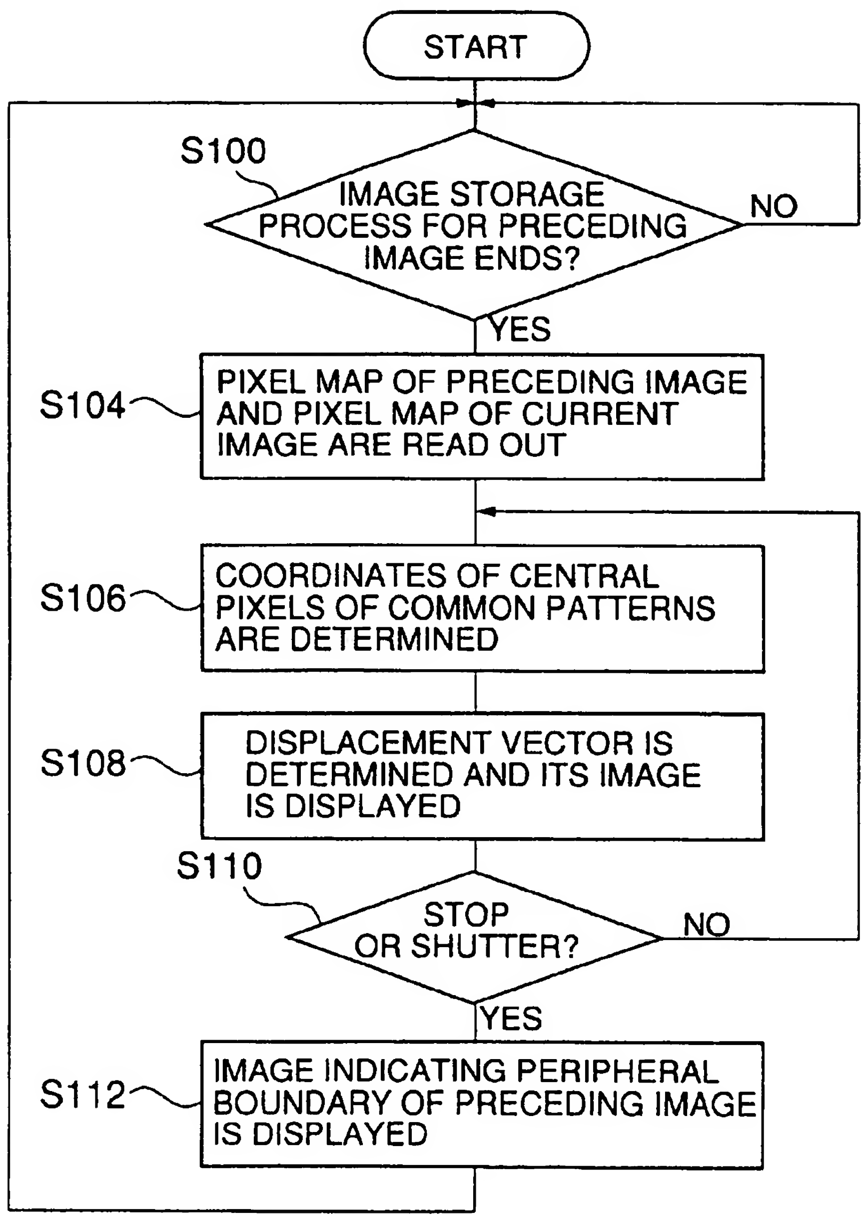

START
S100
IMAGE STORAGE PROCESS FOR PRECEDING IMAGE ENDS?
NO
YES
S104
PIXEL MAP OF PRECEDING IMAGE AND PIXEL MAP OF CURRENT IMAGE ARE READ OUT
S106
COORDINATES OF CENTRAL PIXELS OF COMMON PATTERNS ARE DETERMINED
S108
DISPLACEMENT VECTOR IS DETERMINED AND ITS IMAGE IS DISPLAYED
S110
STOP OR SHUTTER?
NO
YES
S112
IMAGE INDICATING PERIPHERAL BOUNDARY OF PRECEDING IMAGE IS DISPLAYED

FIG.3A

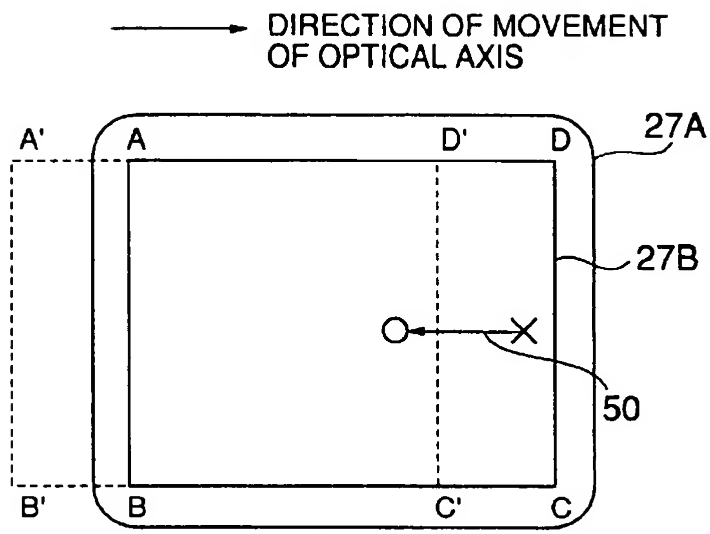

DIRECTION OF MOVEMENT
OF OPTICAL AXIS
A'
A
D'
D
27A
27B
50
B'
B
C'
C

FIG.3B

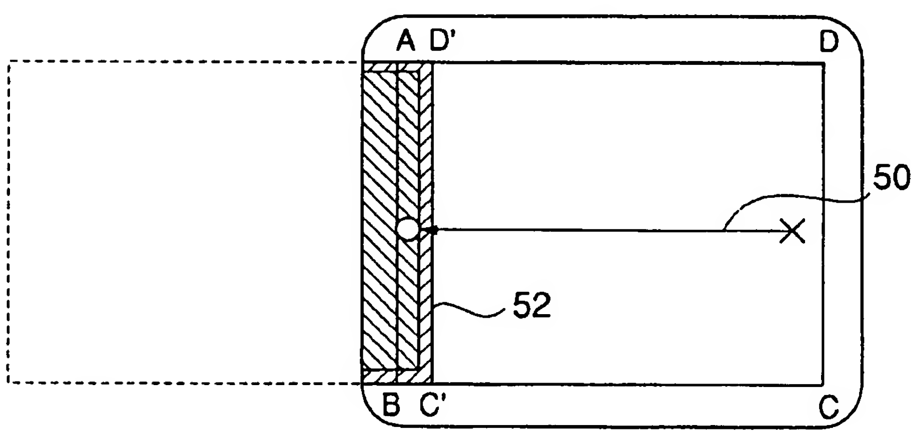

A D'
D
50
52
B C'
C

FIG.4

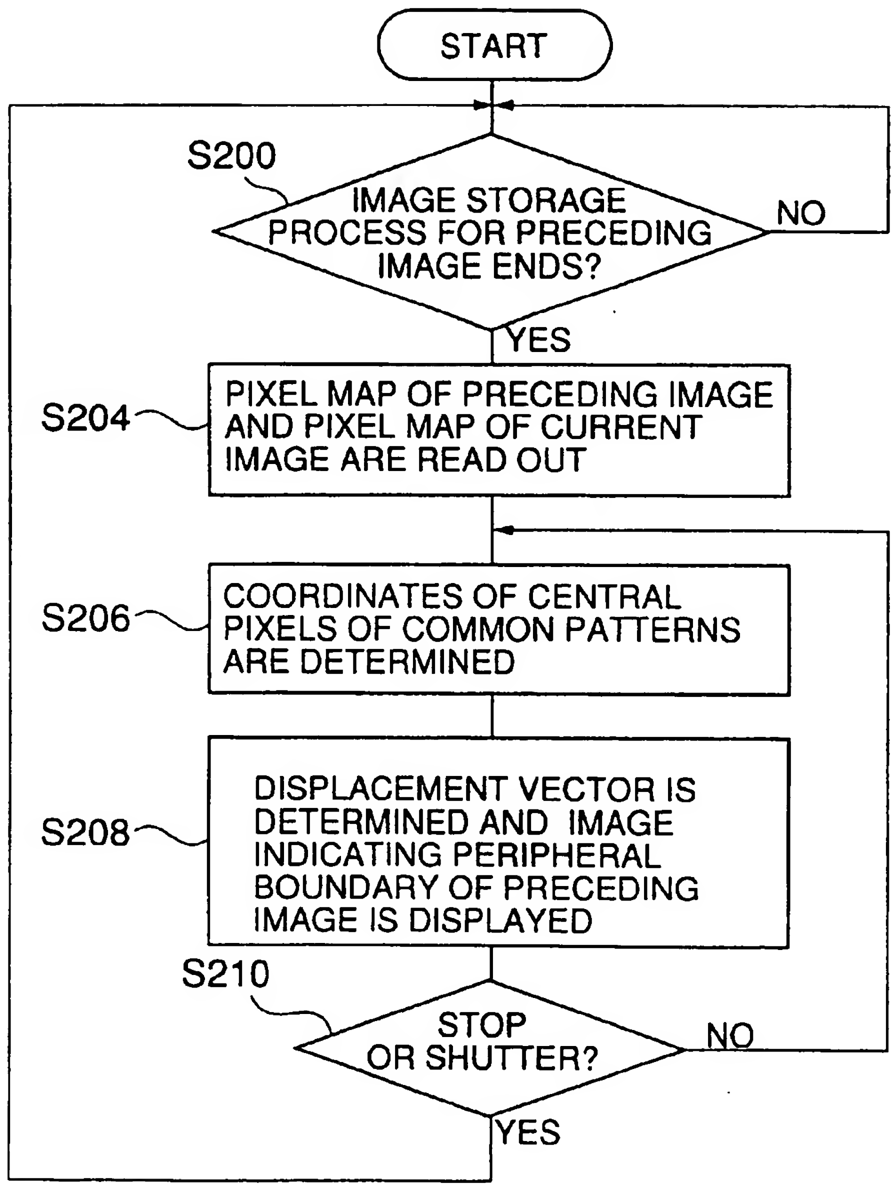

START
S200
IMAGE STORAGE PROCESS FOR PRECEDING IMAGE ENDS?
NO
YES
S204
PIXEL MAP OF PRECEDING IMAGE AND PIXEL MAP OF CURRENT IMAGE ARE READ OUT
S206
COORDINATES OF CENTRAL PIXELS OF COMMON PATTERNS ARE DETERMINED
S208
DISPLACEMENT VECTOR IS DETERMINED AND IMAGE INDICATING PERIPHERAL BOUNDARY OF PRECEDING IMAGE IS DISPLAYED
S210
STOP OR SHUTTER?
NO
YES

FIG.5

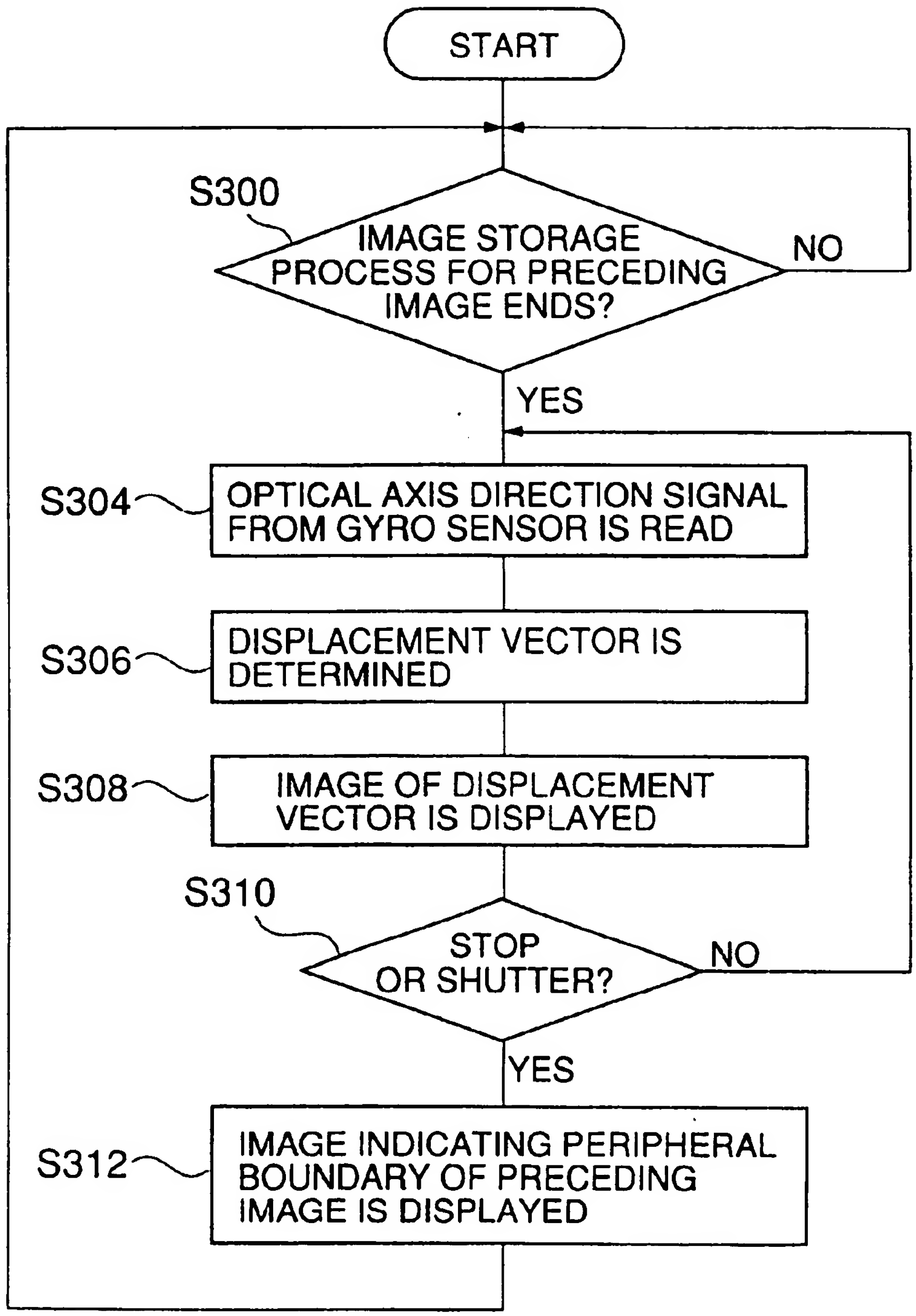

START
S300
IMAGE STORAGE PROCESS FOR PRECEDING IMAGE ENDS?
NO
YES
S304
OPTICAL AXIS DIRECTION SIGNAL FROM GYRO SENSOR IS READ
S306
DISPLACEMENT VECTOR IS DETERMINED
S308
IMAGE OF DISPLACEMENT VECTOR IS DISPLAYED
S310
STOP OR SHUTTER?
NO
YES
S312
IMAGE INDICATING PERIPHERAL BOUNDARY OF PRECEDING IMAGE IS DISPLAYED

FIG.6

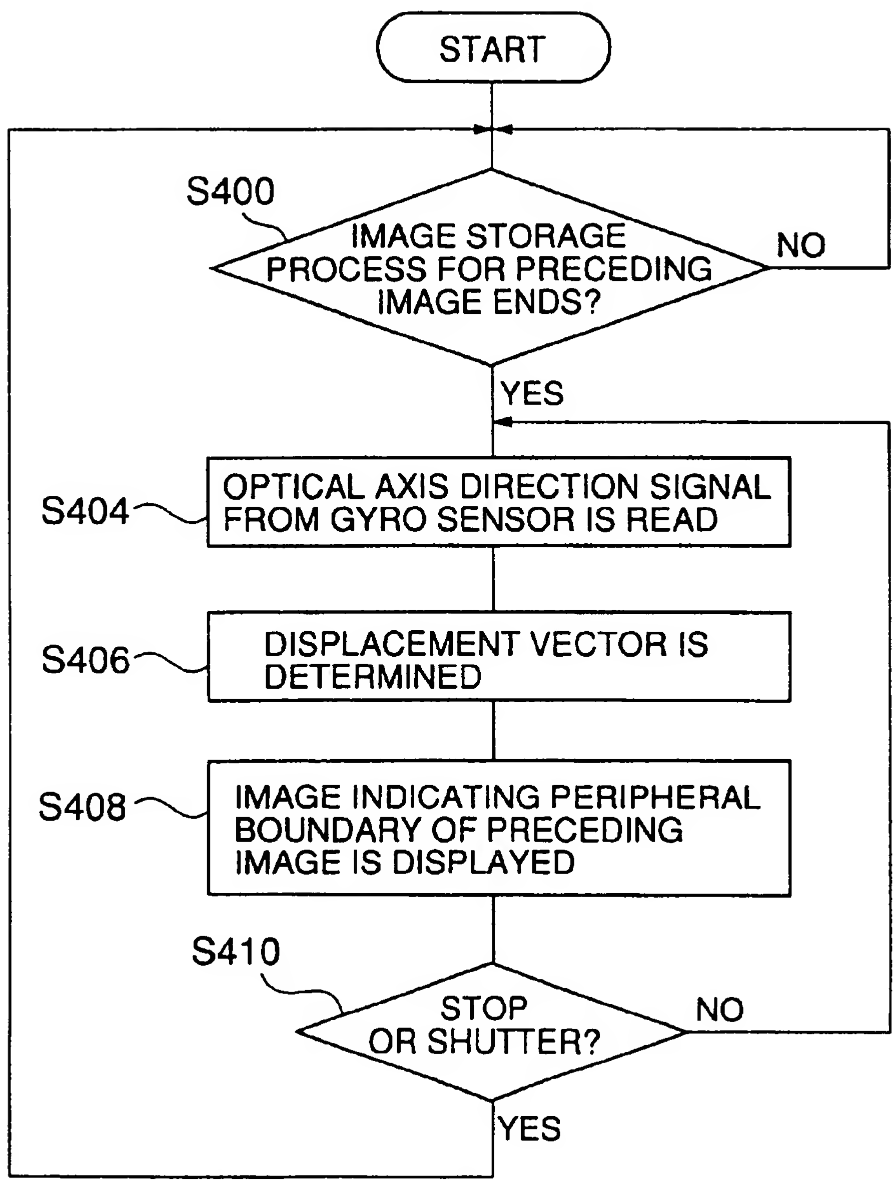

START
S400
IMAGE STORAGE PROCESS FOR PRECEDING IMAGE ENDS?
NO
YES
S404
OPTICAL AXIS DIRECTION SIGNAL FROM GYRO SENSOR IS READ
S406
DISPLACEMENT VECTOR IS DETERMINED
S408
IMAGE INDICATING PERIPHERAL BOUNDARY OF PRECEDING IMAGE IS DISPLAYED
S410
STOP OR SHUTTER?
NO
YES

FIG.7

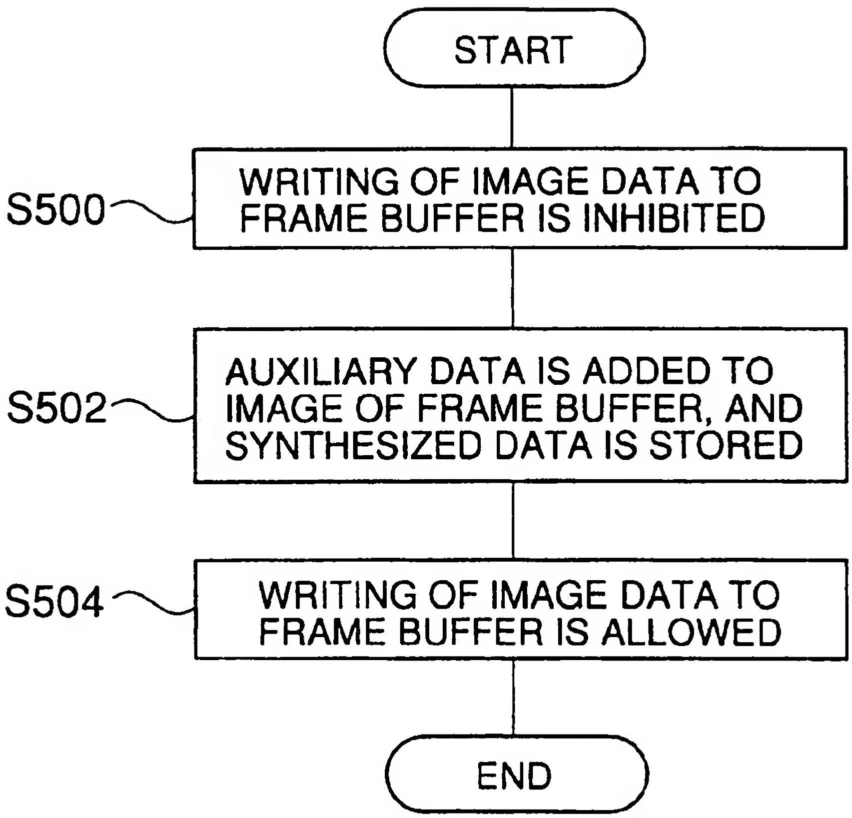

START
S500
WRITING OF IMAGE DATA TO FRAME BUFFER IS INHIBITED
S502
AUXILIARY DATA IS ADDED TO IMAGE OF FRAME BUFFER, AND SYNTHESIZED DATA IS STORED
S504
WRITING OF IMAGE DATA TO FRAME BUFFER IS ALLOWED
END

FIG.8A

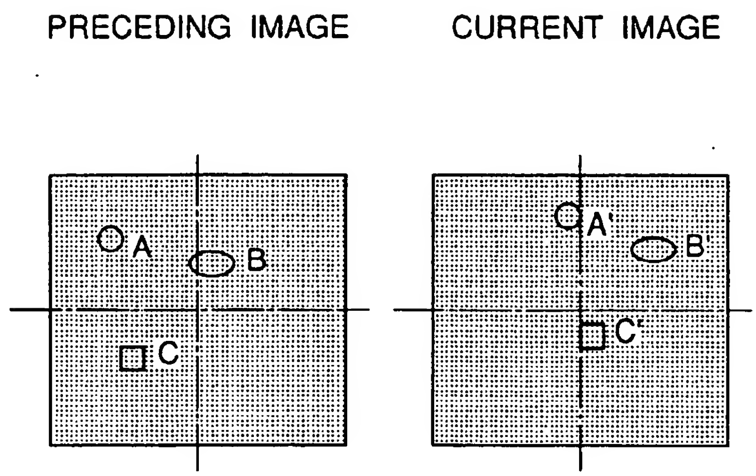
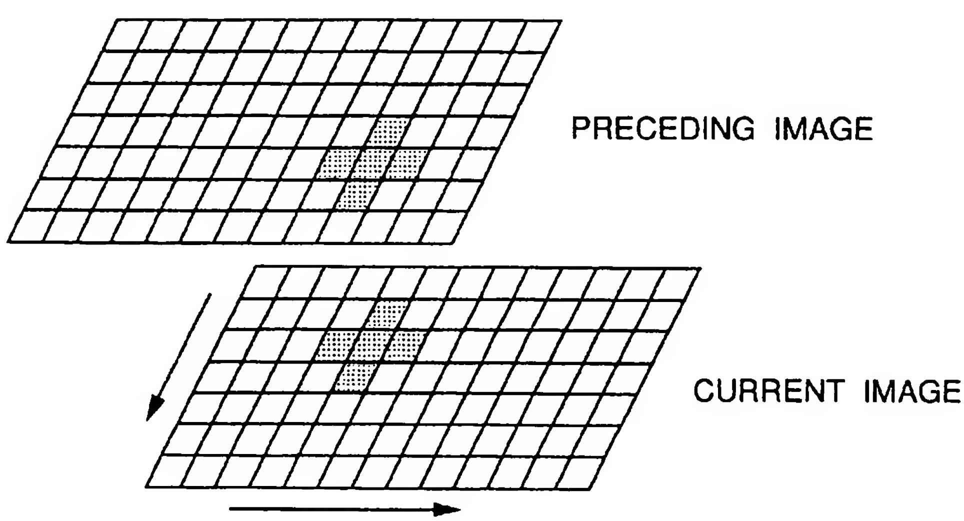

PRECEDING IMAGE
CURRENT IMAGE
A
B
C
A'
B'
C'

FIG.8B


PRECEDING IMAGE
CURRENT IMAGE

SYSTEM AND METHOD FOR DISPLAYING AN IMAGE INDICATING A POSITIONAL RELATION BETWEEN PARTIALLY OVERLAPPING IMAGES

BACKGROUND OF THE INVENTION

(1) Field of the Invention

The present invention relates to a camera system which electronically stores an image of an object and displays the image on a display monitor.

(2) Description of the Related Art

Generally, to achieve an adequately high level of resolution of an image captured by using a digital camera or a video camera, it is necessary to use a zoom-up function of the camera or move the camera close to an object to be imaged. This makes it difficult to obtain an image covering a wide angle related to the object. To capture an image covering a wide angle related to the object, it is necessary to use a zoom-down function of the camera or move the camera away from the object. However, this makes it difficult to obtain an image with a high level of resolution.

In order to obtain a wide-angle image with a high resolution from an object, a divisional shooting method has been proposed. In the divisional shooting method, a plurality of partially overlapping images are successively shot so as to cover a wide angle related to the object, and they are synthesized to create a composite image with an adequate level of resolution.

As disclosed in Japanese Published Utility Model Application No. 8-4783, an image processing device which is capable of combining a plurality of partially overlapping images together to create a composite image is known.

To effectively carry out the divisional shooting method, it is necessary that, after a preceding image is taken and before a current image is taken, the user stop movement of an optical axis of the camera at an appropriate position where an overlapping portion of the two adjacent images is appropriate for subsequently producing a composite image from the images. However, in order to meet this requirement, a conventional digital camera requires a special adapter. If such an adapter is not used, it is difficult for the conventional digital camera to effectively carry out the divisional shooting method. In a case of the conventional digital camera with no special adapter, there is a possibility that no overlapping portion exists between the two adjacent images or a too large overlapping portion be produced between the two adjacent images. If the overlapping images with undesired overlapping portions are obtained through the divisional shooting method, it is difficult to effectively combine or synthesize the images together to create a composite image.

SUMMARY OF THE INVENTION

An object of the present invention is to provide a camera system which displays an image indicating a positional relation among partially overlapping images, and enables an operator to easily and effectively carry out a divisional shooting process.

Another object of the present invention is to provide a divisional shooting method which displays an image indicating a positional relation among partially overlapping images on a screen of a monitor during a divisional shooting mode of a camera system.

The above-mentioned objects of the present invention are achieved by a camera system which comprises: a display monitor which displays an image of an object, taken by an optical unit, on a screen of the monitor; a reading unit which reads a preceding image and a current image among a plurality of partially overlapping images, from a memory device, the preceding image and the current image containing a common element; a determining unit which determines a positional relation between the preceding image and the current image based on a common pattern derived from the common element in the two adjacent images read by the reading unit; and a displaying unit which displays an image indicating a boundary of the preceding image on the screen of the monitor at a shifted position according to the positional relation determined by the determining unit, with the current image concurrently displayed on the screen of the monitor.

The above-mentioned objects of the present invention are achieved by a divisional shooting method for a camera system in which at least two of partially overlapping images of an object, taken by an optical unit, are displayed, comprising the steps of: reading a preceding image and a current image among the partially overlapping images, from a memory device, the preceding image and the current image containing a common element; determining a positional relation between the preceding image and the current image based on a common pattern derived from the common element in the two adjacent images; and displaying an image, indicating a boundary of the preceding image, on a screen of a display monitor at a shifted position according to the positional relation determined by the determining step, with the current image concurrently displayed on the screen of the monitor.

In the camera system of the present invention, a positional relation between the preceding image and the current image is determined based on a common pattern derived from the common element in the two adjacent images. The operator can easily carry out a divisional shooting mode of the camera system by viewing both the current image and the image indicating the positional relation between the partially overlapping images on the screen of the monitor. The positional relation between the preceding image and the current image is clearly noticeable to the operator by viewing the positional relation image on the screen of the monitor together with the current image while the camera is panned in a desired direction. Therefore, the operator easily stops the movement of the optical axis of the camera at an appropriate position by viewing the positional relation image on the screen of the monitor, and turns ON a shutter switch to store the current image.

BRIEF DESCRIPTION OF THE DRAWINGS

Other objects, features and advantages of the present invention will become more apparent from the following detailed description when read in conjunction with the accompanying drawings in which:

FIG. 1 is a block diagram of a preferred embodiment of a camera system of the present invention;

FIG. 2 is a flowchart for explaining a first example of a divisional shooting process performed by a processor of the camera system;

FIG. 3A and FIG. 3B are diagrams showing an image which is displayed on a screen of a display monitor when the camera is moved in a given direction;

FIG. 4 is a flowchart for explaining a second example of the divisional shooting process performed by the processor of the camera system;

FIG. 5 is a flowchart for explaining a third example of the divisional shooting process performed by the processor of the camera system;

FIG. 6 is a flowchart for explaining a fourth example of the divisional shooting process performed by the processor of the camera system;

FIG. 7 is a flowchart for explaining an image storage process performed by the processor of the camera system when a shutter switch is turned ON; and

FIG. 8A and FIG. 8B are diagrams for explaining a determination of a positional relation between partially overlapping images in the divisional shooting process according to the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A description will now be given of the preferred embodiments of the present invention with reference to the accompanying drawings.

In order to carry out a divisional shooting process, the present invention utilizes a method and a system for determining a positional relation between partially overlapping images based upon a common pattern in an overlapping portion of the images. The method and the system are disclosed, for example, in U. S. patent application Ser. No. 08/807,571 filed on Feb. 27, 1997 and U. S. patent application Ser. No. 08/966,889 filed on Nov. 10, 1997, both assigned to the applicant of the present application. The contents of these co-pending applications are hereby incorporated by reference.

FIG. 1 shows a preferred embodiment of a camera system of the present invention. One example of the camera system of the present invention is a digital camera.

As shown in FIG. 1, the camera system of the present embodiment includes an optical unit 10. The optical unit 10 has an image pickup device 12, a lens (not shown), and a lens positioner (not shown). The image pickup device 12 is comprised of a charge-coupled device (CCD). The image pickup device 12 converts light incident from an object into an electrical signal, or an image signal indicative of an input image of the object or the scene. The lens positioner mechanically positions the lens of the optical unit 10 at a desired distance from the object along an optical axis of the lens. Hereinafter, the lens of the optical unit 10 will be referred to as the camera.

In the camera system of the present embodiment, a lens positioner actuator 14 actuates the lens positioner of the optical unit 10 so that the lens is positioned at a desired distance from the object along the optical axis of the lens. An operation part 16 is an operation part of the camera system of FIG. 1, which includes a mode selection switch 18, a shutter switch 20, and other control switches (not shown). An operator can manipulate one of such switches of the operation part 16 so as to select one of operational modes of the camera system or to release the shutter of the camera system.

In the camera system of the present embodiment, a video control unit 24 converts the signal from the image pickup device 12 into a digital signal, processes the digital signal to produce a frame of the input image, and stores the frame in a frame buffer 25. The frame or image defined in the frame buffer 25 is a pixel map that has an array of pixel data, each indicating an intensity (and/or a color value) for a position of a corresponding one of the picture elements, or pixels, in the image. The video control unit 24 displays the image defined in the frame buffer 25 on a liquid-crystal display (LCD) monitor 27, accessing the frame buffer 25 as frequently as a scan rate of the monitor 27. The monitor 27 has a display screen 27A, and the image defined in the frame buffer 25 is displayed on the screen 27A of the monitor 27 by the video control unit 24.

The video control unit 24 further includes a frame buffer 26 in addition to the frame buffer 25. The frame buffer 26 stores auxiliary data indicative of a peripheral boundary 27B (which will be described later) corresponding to the image defined in the frame buffer 25. The video control unit 24 displays the peripheral boundary 27B, indicated by the auxiliary data defined in the frame buffer 26, on the screen 27A of the monitor 27, accessing the frame buffer 26 at the same time as the frame buffer 25. Hence, the image defined in the frame buffer 25 and the auxiliary data defined in the frame buffer 26 are synthesized so that the image with the peripheral boundary 27B is displayed on the screen 27A of the monitor 27 in an overlaid manner. The auxiliary data defined in the frame buffer 26 includes a frame number to identify a captured image among a plurality of partially overlapping images, which will be described later. Further, the auxiliary data may further include image data of a displacement vector or a direction of the optical axis of the camera, which will be described later.

In the camera system of the present embodiment, an image memory 28 is a storage device which stores an image captured by the video control unit 24. The image memory 28 may be any image storage device, for example, one of semiconductor memories including flash memories, or one of magnetic disks including floppy disks and mini-disks (MD).

In the camera system of the present embodiment, a processor 30 controls the overall operation of the camera system and carries out a divisional shooting process including determination of a positional relation between partially overlapping images based upon a common pattern in an overlapping portion of the images. The processor 30 includes an arithmetic control unit 32, a read-only memory (ROM) 33, and a random access memory (RAM) 36. The ROM 33 stores a number of programs 34A through 34N, and fixed information, such as character fonts. The arithmetic control unit 32 carries out individual control operations for the elements of the camera system when one of the programs 34A through 34N in the ROM 33 is executed by the processor 30. The RAM 36 is a main memory of the processor 30 which is available to any of the programs when it is executed. The RAM 36 serves as a work memory available to the arithmetic control unit 32. Further, the processor 30 includes a power supply circuit (not shown) which supplies power to the camera system, and an interface (not shown) which connects the camera system with an external host computer.

In the camera system of FIG. 1, the operator can select one of the operational modes by using the mode selection switch 16. In the present embodiment, the operational modes of the camera system include a normal shooting mode and a divisional shooting mode.

When the normal shooting mode is selected by the mode selection switch 16, a single image of an object or a scene is captured through the image pickup device 12, the image displayed on the screen 27A of the monitor 27 is viewed, and the shutter switch 20 is turned ON by the operator so that the image defined in the frame memory 25 is stored in the image memory 28.

When the divisional shooting mode is selected in the camera system of the present embodiment, a plurality of partially overlapping images are successively shot so as to cover a wide angle related to an object to be imaged, and they are synthesized to create a composite image with an

adequate level of resolution. The divisional shooting mode is useful to obtain a panoramic image or a high-resolution image through image composition. The camera system of the present invention is particularly relevant to the divisional shooting mode, and the following description will be given of an operation of the camera system of the present embodiment when the divisional shooting mode is performed.

In the camera system of the present embodiment, when the divisional shooting mode is selected by the mode selection switch 20, the processor 30 starts the execution of a divisional shooting processing program 34I among the programs 34A through 34N in the ROM 33. A divisional shooting process is performed by the processor 30 according to the divisional shooting processing program 34I.

In order to take a first one of partially overlapping images when the divisional shooting process is started, the operator directs the optical axis of the camera (or the lens of the optical unit 10) to an object to be imaged. In accordance with the signal from the image pickup device 12, the video control unit 24 stores a corresponding frame in the frame memory 25, and displays the image on the screen 27A of the monitor 27. The operator turns ON the shutter switch 20 of the operation part 16 while viewing the image on the screen 27A of the monitor 27. A shutter signal from the operation part 16 is sent to the processor 30 immediately after the shutter switch 20 is turned ON. In response to the shutter signal, the processor 30 stores the image, defined in the frame memory 25 of the video control unit 24, in the image memory 28.

The above-mentioned image storage process is performed by the processor 30 of the camera system in accordance with an image storage processing program 34N among the programs 34A through 34N stored in the ROM 33. The execution of the image storage processing program 34N is started by the processor 30 in response to the shutter signal. During the image storage process, all the image data corresponding to the entire screen 27A of the monitor 27 is not stored in the image memory 28, but only a portion of the image data corresponding to an internal portion of the screen 27A of the monitor 27 within the peripheral boundary 27B is stored in the image memory 28. The processor 30 adds a frame number to the auxiliary data of the frame buffer 26 and stores such data defined in the frame buffer 26, in the image memory 28, together with the image defined in the frame buffer 25, during the image storage process. The data being stored in the image memory 28 may be compressed in a compact form or may not be compressed in the original form. During the image storage process, the writing of image data to the frame buffer 25 is inhibited and the image displayed on the screen 27A of the monitor 27 is fixed. Before the image storage process ends, the writing of image data to the frame buffer 25 is allowed. Hence, after the image storage process is performed, the image defined in the frame buffer 25 can be variably updated according to the movement of the optical axis of the camera, and the resulting image is displayed on the screen 27A of the monitor 27.

FIG. 7 shows an image storage process performed by the processor 30 of the camera system of the present embodiment. The image storage processing program 34N among the programs 34A through 34N in the ROM 33 is loaded to the RAM 36 and executed by the processor 30 immediately after the shutter switch 20 is turned ON by the operator. Then, the image storage process of FIG. 7 is performed by the processor 30 according to the image storage processing program 34N.

As shown in FIG. 7, at the start of the image storage process, the processor 30 at step S500 inhibits the writing of image data to the frame buffer 25 by the video control unit 24. Hence, during the image storage process, the image displayed on the screen 27A of the monitor 27 is fixed.

The processor 30 at step S502 combines the auxiliary data of the frame buffer 26 with the image of the frame buffer 25 to create a synthesized image, and stores the synthesized image in the image memory 28. As described above, the auxiliary data of the frame buffer 26 includes a frame number to identify a captured image among the partially overlapping images. The auxiliary data of the frame buffer 26 may include other parameter values (which will be described later). However, when the image storage process with respect to a first one of partially overlapping images is performed, the auxiliary data of the frame buffer 26 is null or vacant, and only the image of the frame buffer 25 is stored in the image memory 28 at the step S502.

The processor 30 at step S504 allows the writing of image data to the frame buffer 25 by the video control unit 24. After the step S504 is performed, the image storage process of FIG. 7 ends. Hence, after the image storage process is performed, the image defined in the frame buffer 25 is displayed on the screen 27A of the monitor 27.

After the first one of the partially overlapping images is taken, the operator pans the camera in a desired direction in order to take a following one of the partially overlapping images during the divisional shooting mode. By viewing the preceding image with the peripheral boundary on the screen 27A of the monitor 27, the operator stops the movement of the optical axis of the camera at an appropriate position where an overlapping portion of the two adjacent images is appropriate for subsequently producing a composite image from the images. Then, the current image is captured and stored in the image memory 28 in a similar manner. The above-described procedure is repeated until all the partially overlapping images for the object to be imaged are captured and stored. In this manner, the partially overlapping images are successively shot so as to cover a wide angle related to the object, and they are synthesized to create a composite image with an adequate level of resolution by using the technology as disclosed in the above-mentioned U. S. patent applications.

According to the camera system of the present invention, the operator can easily carry out the divisional shooting process by viewing both the current image and the peripheral boundary 27B (or the preceding image) on the screen 27A of the monitor 27. A positional relation between the preceding image and the current image is clearly noticeable to the operator by viewing the peripheral boundary 27B on the screen 27A of the monitor 27 and the current image while the camera is panned in the desired direction. Therefore, the operator easily stops the movement of the optical axis of the camera at an appropriate position by viewing an image of the peripheral boundary 27B, and turns ON the shutter switch 20 to store the current image.

FIG. 2 shows a first example of the divisional shooting process performed by the processor 30 in accordance with the divisional shooting processing program 34I.

As shown in FIG. 2, at the start of the divisional shooting process, the processor 30 at step S100 detects whether the image storage process, shown in FIG. 7, with respect to a preceding one of the partially overlapping images ends. The end of the image storage process is notified to the arithmetic control unit 32 when the execution of the image storage processing program 34N has normally ended. When the result at the step S100 is negative, the processor 30 repeats the step S100.

When the result at the step S100 is affirmative, the processor 30 at step S104 reads out the pixel map of the preceding image from the image memory 28, and reads out the pixel map of a currently-captured image from the frame buffer 25. These pixel maps are temporarily stored in the RAM 36. The pixel map of the preceding image is selected as a standard image. Each of the pixel data of the two adjacent images corresponding to an overlapping portion of the images is divided into blocks of a predetermined size, for example, 16 by 16 pixels.

After the step S104 is performed, the processor 30 at step S106 performs a matching between corresponding blocks from an overlapping portion of the two adjacent images. During the step S106, a common pattern in the two adjacent images is identified if a certain similarity threshold is met. This matching may be performed by checking the intensities of individual pixels of the corresponding blocks. This is useful for reducing the amount of required calculations. Alternatively, the matching may be performed by checking the color values of individual pixels of the corresponding blocks, but this will increase the amount of required calculations. The above matching procedures are repeated until all the blocks are processed so that a maximum-similarity common pattern in the preceding image and the maximum-similarity common pattern in the current image are detected.

A method and a system for determining a positional relation between partially overlapping images based upon a common pattern in an overlapping portion of the images are disclosed in the above-mentioned U. S. patent applications, and the divisional shooting process according to the present invention utilizes the method and the system.

As previously described, during the step S106 of the divisional shooting process of FIG. 2, a determination of a positional relation between partially overlapping images is carried out. By referring to FIG. 8A and FIG. 8B, a detailed procedure of the determination of the positional relation in the step S106 will now be described.

It is supposed that the pixel map of the preceding image from the image memory 28 and the pixel map of the current image from the frame buffer 25 have been read out as in the step S104. These pixel maps are temporarily stored in the RAM 36. The pixel map of the preceding image is selected as the standard image. Each of the two adjacent images corresponding to an overlapping portion of the images is divided into blocks of a predetermined size.

As shown in FIG. 8A, pixels "A", "B" and "C" in the preceding image and pixels "A'", "B'" and "C'" in the current image correspond to the overlapping portion of the images. During the step S106, a matching between corresponding blocks from the overlapping portion of the two adjacent images is performed. A common pattern (such as the pixels A, B and C and the pixels A', B' and C') in the two adjacent images is identified if a certain similarity threshold is met. This matching may be performed by checking the intensities of individual pixels of the corresponding blocks. The above matching procedures are repeated until all the blocks are processed, so that a maximum-similarity common pattern in the preceding image and the maximum-similarity common pattern in the current image are detected.

As shown in FIG. 8B, the maximum-similarity common pattern in the two images is detected if the difference between the pixel values (or the intensities of the pixels A and A', the pixels B and B' or the pixels C and C') of the corresponding blocks is found to be the minimum when the current image is moved relative to the preceding image by both a distance for a first number of pixels in the x-axis direction and a distance for a second number of pixels in the y-axis direction. Through the above pixel-based method, the processor 30 detects the maximum-similarity common pattern in the two images. That is, the processor 30 at the step S106 carries out the determination of the positional relation between the partially overlapping images.

In the above-described procedure, the maximum-similarity common pattern in the two images is detected by using the pixel-based method, in order to carry out the determination of the positional relation between the partially overlapping images. However, according to the present invention, it is also possible to achieve the determination of a positional relation between partially overlapping images at an accuracy higher than the accuracy of one pixel. As previously described, the determination of a positional relation between partially overlapping images based upon a common pattern in an overlapping portion of the images are disclosed in the above-mentioned U. S. patent applications, and, for that purpose, the divisional shooting process according to the present invention may utilize the method and the system.

Referring back to FIG. 2, during the step S106, the processor 30 further determines both coordinates (I, J) of a central pixel of the maximum-similarity common pattern in the preceding image and coordinates (Im, Jm) of a central pixel of the maximum-similarity common pattern in the current image. The coordinates (I, J) and the coordinates (Im, Jm) based on a screen coordinate system of the screen 27A of the monitor 27 are determined by the processor 30.

The processor 30 at step S108 determines a displacement vector (I-Im, J-Jm), which indicates a positional relation between the preceding image and the current image, by the difference between the coordinates (I, J) and the coordinates (Im, Jm). In the step S108, after the contents of the frame buffer 26 are cleared, the processor 30 writes image data, indicative of the displacement vector, to the frame buffer 26 as part of the auxiliary data. Hence, the image of the displacement vector (or the auxiliary data defined in the frame buffer 26) is displayed on the screen 27A of the monitor 27.

The processor 30 at step S110 detects whether the operator stops the movement of the optical axis of the camera (or detects whether the operator turns ON the shutter switch 20). When the result at the step S110 is negative, the above steps S106 and S108 are repeated.

When the step S106 is performed for second or subsequent ones of the partially overlapping images, the coordinates (I, J) of the central pixel of the maximum-similarity common pattern in the preceding image and the direction of the displacement vector are known. The matching procedures in the step S106 may be performed for only the blocks of the current image in the overlapping portion of the two images, indicated by the direction of the displacement vector and the coordinates (I, J). By using such a simplified matching, the common pattern in the two adjacent images may be identified, and coordinates (Im, Jm) of the central pixel of the maximum-similarity common pattern in the current image may be determined.

The operator stops the panning of the camera at an appropriate position where an appropriate overlapping portion of the two adjacent images can be seen with the image of the displacement vector on the screen 27A of the monitor 27, and turns ON the shutter switch 20 to store the current image. Every time the steps S106 and S108 are performed, the processor 30 compares the currently obtained displacement vector and the previously obtained displacement vector

(stored in an internal register of the processor **30** or the RAM **36**) so as to determine whether the operator stops the movement of the optical axis of the camera. If the difference between the two displacement vectors is larger than a threshold value, the result at the step **S110** is negative. If the difference between the two displacement vectors is less than the threshold value, the result at the step **S110** is affirmative.

When the result at the step **S110** is affirmative, the processor **30** at step **S112** writes image data, indicative of the peripheral boundary **27B** of the preceding image, to the frame buffer **26** at a position shifted from the previous position. The shifted position is determined from the previous position based on the magnitude and direction of the displacement vector obtained in the step **S108**. Hence, the image of the peripheral boundary **27B** defined in the frame buffer **26** is displayed on the screen **27A** of the monitor **27** as if the peripheral boundary **27B** is shifted according to the movement of the optical axis of the camera.

In the step **S112**, the image data of the displacement vector obtained in the step **S108** may be left in the frame buffer **26** without change. Alternatively, the image data of the displacement vector in the frame buffer **26** may be deleted, and then the image data of the shifted peripheral boundary **27B** may be defined in the frame buffer **26**. The image of the peripheral boundary **27B** being displayed on the screen **27A** of the monitor **27** may be a frame of the preceding image or a solid model of the preceding image with a certain color attached to the internal pixels.

The operator can easily carry out the divisional shooting process with the camera system by viewing both the current image and the peripheral boundary **27B** (or the preceding image) on the screen **27A** of the monitor **27**. A positional relation between the preceding image and the current image is clearly noticeable to the operator by viewing the peripheral boundary **27B** on the screen **27A** of the monitor **27** and the current image while the camera is panned in a desired direction. Therefore, the operator easily stops the movement of the optical axis of the camera at an appropriate position by viewing an image of the peripheral boundary **27A**, and turns ON the shutter switch **20** to store the current image.

After the step **S112** is performed, the control is transferred to the step **S100**. The processor **30** at the step **S100** waits for the end of the image storage process at which the currently captured image is further stored in the image memory **28**. As described above, during the image storage process, the frame number for the current image and the displacement vector for the current image are added to the auxiliary data of the frame buffer **26** and such data defined in the frame buffer **26** is stored in the image memory **28** together with the image defined in the frame buffer **25**. The frame number and the displacement data are used when synthesizing the partially overlapping images to create a composite image.

FIG. **3A** shows an image which is displayed on the screen **27A** of the monitor **27** when the camera is being moved in a given direction indicated in FIG. **3A**. In FIG. **3A**, a peripheral boundary of a preceding image is indicated by the dotted-line rectangle A'B'C'D', and a peripheral boundary of a current image is indicated by the solid-line rectangle ABCD. A displacement between the preceding image and the current image proportional to the movement of the optical axis of the camera is defined by the displacement vector. In the case of FIG. **3A**, the displacement vector is directed to the left and has a length proportional to the movement of the optical axis of the camera. An image **50** of the displacement vector is displayed on the screen **27A** of the monitor **27** as indicated in FIG. **3A**. Although the contents of the preceding image are not displayed, the operator can easily notice a positional relation between the preceding image and the current image on the screen **27A** of the monitor **27** with the image **50**.

FIG. **3B** shows an image which is displayed on the screen **27A** of the monitor **27** when the movement of the optical axis of the camera is stopped and the shutter switch **20** is turned ON by the operator. In FIG. **3B**, an image **52** of the peripheral boundary **27B**, which is displayed on the screen **27A** of the monitor **27**, is indicated by the rectangle ABC'D'. The rectangle ABC'D' corresponds to an overlapping portion of the two adjacent images. As described above, the image data, indicative of the peripheral boundary **27B** of the preceding image, is written to the frame buffer **26** at positions shifted from the previous positions according to the movement of the optical axis of the camera. The image **50** of the displacement vector corresponding to the magnitude and direction of the displacement vector is displayed on the screen **27A** of the monitor **27**. The operator can clearly notice an appropriate overlapping portion of the two images by the image **50** of the displacement vector and the image **52** of the peripheral boundary **27B**. The image **50** of the displacement vector, at the time the movement of the optical axis of the camera is stopped, may be displayed on the screen **27A** of the monitor **27**. Alternatively, the display of the image **50** of the displacement vector may be omitted.

FIG.**4** shows a second example of the divisional shooting process performed by the processor **30** in accordance with the divisional shooting processing program **34I**.

As shown in FIG. **4**, at the start of the divisional shooting process in the present embodiment, the processor **30** at step **S200** detects whether the image storage process with respect to a preceding one of the partially overlapping images ends. The end of the image storage process is notified to the arithmetic control unit **32** when the execution of the image storage processing program **34N** has normally ended. When the result at the step **S200** is negative, the processor **30** repeats the step **S200**.

When the result at the step **S200** is affirmative, the processor **30** at step **S204** reads out the pixel map of the preceding image from the image memory **28**, and reads out the pixel map of the currently-captured image from the frame buffer **25**. The pixel maps are temporarily stored in the RAM **36**. The pixel map of the preceding image is selected as a standard image. Each of the pixel data of the two adjacent images corresponding to the overlapping portion of the images is divided into blocks of a predetermined size, for example, 16 by 16 pixels.

After the step **S204** is performed, the processor **30** at step **S206** performs a matching between corresponding blocks from the two adjacent images. During the step **S206**, a common pattern in the two adjacent images is identified if a certain similarity threshold is met. The matching procedures are repeated for every block until all the blocks are processed so that the common pattern in the preceding image and the common pattern in the current image are identified.

Further, during the step **S206**, the processor **30** determines both coordinates (I, J) of a central pixel of a maximum-similarity common pattern in the preceding image and coordinates (Im, Jm) of a central pixel of the maximum-similarity common pattern in the current image. The coordinates (I, J) and the coordinates (Im, Jm) based on a screen coordinate system of the screen **27A** of the monitor **27** are determined by the processor **30**.

The steps **S200–S206** in the present embodiment are essentially the same as the steps **S100–S106** in the embodiment of FIG. **2**.

The processor at step S208 determines a displacement vector (I-Im, J-Jm), which indicates a positional relation between the preceding image and the current image, by the difference between the coordinates (I, J) and the coordinates (Im, Jm). In the present embodiment, during the step S208, the processor 30 writes image data, indicative of the peripheral boundary 27B of the preceding image, to the frame buffer 26 at positions shifted from the previous positions. The shifted positions are indicated by the magnitude and direction of the displacement vector. Hence, the image of the peripheral boundary 27B defined in the frame buffer 26 is displayed on the screen 27A of the monitor 27.

Unlike the embodiment of FIG. 2, during the step S208 in the present embodiment, the processor 30 does not write the image data of the displacement vector to the frame buffer 26 as part of the auxiliary data. Hence, the image of the displacement vector is not displayed on the screen 27A of the monitor 27.

The processor 30 at step S210 detects whether the operator stops the movement of the optical axis of the camera (or detects whether the operator turns ON the shutter switch 20). When the result at the step S210 is negative, the above steps S206 and S208 are repeated.

The operator stops the panning of the camera at an appropriate position where an appropriate overlapping portion of the two adjacent images can be seen with the image of the displacement vector on the screen 27A of the monitor 27, and turns ON the shutter switch 20 to store the current image. Every time the steps S206 and S208 are performed, the processor 30 compares the currently obtained displacement vector and the previously obtained displacement vector (stored in the internal register or the RAM 36) so as to determine whether the operator stops the panning of the camera. If the difference between the two displacement vectors is larger than a threshold value, the result at the step S210 is negative. If the difference between the two displacement vectors is less than the threshold value, the result at the step S210 is affirmative.

When the result at the step S210 is affirmative, the control is transferred to the step S200. The processor 30 at the step S200 waits for the end of the image storage process at which the currently captured image is further stored in the image memory 28.

In the present embodiment, the operator can view a peripheral boundary image indicating a positional relation between the current image and the preceding image before the movement of the optical axis of the camera is stopped or the shutter switch 20 is turned ON. The operator can easily carry out the divisional shooting process with the camera system, but the current image and the peripheral boundary image are always displayed on the screen 27A of the monitor 27. It is desirable that the intensity and/or color of the peripheral boundary image may be set at a suitable value so as to prevent the peripheral boundary image from hindering the check for the current image on the screen 27A of the monitor 27.

FIG. 5 shows a third example of the divisional shooting process performed by the processor 30 in accordance with the divisional shooting processing program 34I.

In the present embodiment, the camera system further includes a three-dimensional gyro sensor 40 connected to the arithmetic control unit 32 of the processor 30 as indicated by the dotted line in FIG. 1. The sensor 40 detects a three-dimensional direction of the optical axis of the optical unit 10 and outputs a signal indicating the optical axis direction to the arithmetic control unit 32 of the processor 30. The sensor 40 may be a built-in type or an external-installation type for the camera system. Other elements of the camera system in the present embodiment are the same as corresponding elements of the camera system shown in FIG. 1, and a description thereof will be omitted.

When the divisional shooting mode is selected by the mode selection switch 20, the processor 30 starts the execution of the divisional shooting processing program 34I in the ROM 33. The present embodiment of the divisional shooting process is performed by the processor 30 according to the divisional shooting processing program 34I.

In order to take a first one of partially overlapping images at the start of the divisional shooting process is started, the operator directs the optical axis of the camera to an object to be imaged and turns ON the shutter switch 20. A shutter signal from the operation part 16 is sent to the processor 30 immediately after the shutter switch 20 is turned ON. In response to the shutter signal, the processor 30 reads a signal output by the sensor 40 at that time, and temporarily stores the signal in an internal register of the processor 30 or the RAM 36. In accordance with the signal from the image pickup device 12, the video control unit 24 stores a corresponding frame in the frame memory 25, and displays the image on the screen 27A of the monitor 27. In response to the shutter signal, the processor 30 stores the image, defined in the frame memory 25, in the image memory 28.

The above-mentioned image storage process is performed by the processor 30 according to the image storage processing program 34N in the ROM 33. The execution of the image storage processing program 34N is started by the processor 30 in response to the shutter signal. During the image storage process, the processor 30 adds both the frame number and the optical axis direction signal to the auxiliary data of the frame buffer 26, and stores such data defined in the frame buffer 26, in the image memory 28, together with the image defined in the frame buffer 25. During the image storage process, the writing of image data to the frame buffer 25 is inhibited and the image displayed on the screen 27A of the monitor 27 is fixed. Before the image storage process ends, the writing of image data to the frame buffer 25 is allowed. Hence, after the image storage process is performed, the image defined in the frame buffer 25 can be variably updated according to the movement of the optical axis of the camera, and the resulting image is displayed on the screen 27A of the monitor 27.

After the first one of the partially overlapping images is taken, the operator pans the camera in a desired direction in order to take a following one of the partially overlapping images during the divisional shooting mode. By viewing the preceding image with the peripheral boundary on the screen 27A of the monitor 27, the operator stops the movement of the optical axis of the camera such that the preceding image and the currently-captured image overlap each other with an appropriate overlapping portion of the images. Then, the current image is captured and stored in the image memory 28 together with the auxiliary data, including the frame number and the optical axis direction signal, in a similar manner. The above-described procedure is repeated until all the partially overlapping images for the object to be imaged are captured and stored.

With reference to FIG. 5, a description will now be given of the third example of the divisional shooting process performed by the processor 30.

As shown in FIG. 5, at the start of the divisional shooting process in the present embodiment, the processor 30 at step S300 detects whether the image storage process of FIG. 7

with respect to a preceding one of the partially overlapping images ends. The end of the image storage process is notified to the processor 30. When the result at the step S300 is negative, the processor 30 repeats the step S300.

When the result at the step S300 is affirmative, the processor 30 at step S304 reads an optical axis direction signal (related to the current image) output by the sensor 40 at that time, and reads the optical axis direction signal (related to the preceding image) from the internal register or the RAM 36.

After the step S304 is performed, the processor 30 at step S306 determines a displacement vector, which indicates a positional relation of the preceding image to the current image on the screen 27A of the monitor 27, by the difference between the optical axis direction signal related to the preceding image and the optical axis direction signal related to the current image.

The processor 30 at step S308 writes image data, indicative of the displacement vector, to the frame buffer 26 as part of the auxiliary data after the contents of the frame buffer 26 are cleared. Hence, an image of the displacement vector (or the auxiliary data defined in the frame buffer 26) is displayed on the screen 27A of the monitor 27, similar to the image 50 shown in FIG. 3A and FIG. 3B.

The processor 30 at step S310 detects whether the operator stops the movement of the optical axis of the camera (or detects whether the operator turns ON the shutter switch 20). When the result at the step S310 is negative, the above steps S304 through S308 are repeated.

The operator stops the panning of the camera at an appropriate position where an appropriate overlapping portion of the two adjacent images can be seen with the image of the displacement vector on the screen 27A of the monitor 27, and turns ON the shutter switch 20 to store the current image. Every time the steps S304 through S308 are performed, the processor 30 compares the currently obtained displacement vector and the previously obtained displacement vector (stored in the internal register or the RAM 36) so as to determine whether the operator stops the movement of the optical axis of the camera. If the difference between the two displacement vectors is larger than a threshold value, the result at the step S310 is negative. If the difference between the two displacement vectors is less than the threshold value, the result at the step S310 is affirmative.

When the result at the step S310 is affirmative, the processor 30 at step S312 writes image data, indicative of the peripheral boundary 27B of the preceding image, to the frame buffer 26 at positions shifted from the previous positions. The shifted positions are indicated by the magnitude and direction of the displacement vector obtained in the step S306. Hence, the image of the peripheral boundary 27B defined in the frame buffer 26 is displayed on the screen 27A of the monitor 27.

In the step S312, the image data of the displacement vector obtained in the step S306 may be left in the frame buffer 26 without change. Alternatively, the image data of the displacement vector in the frame buffer 26 may be deleted, and then the image data of the shifted peripheral boundary 27B may be defined in the frame buffer 26. The image of the peripheral boundary 27B being displayed on the screen 27A of the monitor 27 may be a frame of the preceding image or a solid model of the preceding image with a certain color attached to the internal pixels.

The operator can easily carry out the divisional shooting process with the camera system by viewing both the current image and the peripheral boundary 27B (or the preceding image) on the screen 27A of the monitor 27. A positional relation between the preceding image and the current image is clearly noticeable to the operator by viewing the peripheral boundary 27B on the screen 27A of the monitor 27 and the current image while the camera is panned in a desired direction. Therefore, the operator easily stops the movement of the optical axis of the camera at an appropriate position by viewing an image of the peripheral boundary 27B, and turns ON the shutter switch 20 to store the current image.

After the step S312 is performed, the control is transferred to the step S300. The processor 30 at the step S300 waits for the end of the image storage process at which the currently captured image is further stored in the image memory 28. As described above, during the image storage process, the frame number for the current image and the displacement vector for the current image are added to the auxiliary data of the frame buffer 26 and such data defined in the frame buffer 26 is stored in the image memory 28 together with the image defined in the frame buffer 25. The frame number and the displacement data are used when synthesizing the partially overlapping images to create a composite image.

FIG. 6 shows a fourth example of the divisional shooting process performed by the processor 30 in accordance with a divisional shooting processing program 34I.

As shown in FIG. 6, at the start of the divisional shooting process in the present embodiment, the processor 30 at step S400 detects whether the image storage process of FIG. 7 with respect to a preceding one of the partially overlapping images ends. The end of the image storage process is notified to the processor 30. When the result at the step S400 is negative, the processor 30 repeats the step S400.

When the result at the step S400 is affirmative, the processor 30 at step S404 reads an optical axis direction signal (related to the current image) output by the sensor 40 at that time, and reads the optical axis direction signal (related to the preceding image) from the internal register or the RAM 36.

After the step S404 is performed, the processor 30 at step S406 determines a displacement vector, which indicates a positional relation of the preceding image to the current image on the screen 27A of the monitor 27, by the difference between the optical axis direction signal related to the preceding image and the optical axis direction signal related to the current image.

The processor 30 at step S408 writes image data, indicative of the peripheral boundary 27B of the preceding image, to the frame buffer 26 at positions shifted from the previous positions. The shifted positions are indicated by the magnitude and direction of the displacement vector obtained in the step S406. Hence, an image of the peripheral boundary 27B defined in the frame buffer 26 is displayed on the screen 27A of the monitor 27, similar to the image 52 shown in FIG. 3B.

The processor 30 at step S410 detects whether the operator stops the movement of the optical axis of the camera (or detects whether the operator turns ON the shutter switch 20). When the result at the step S410 is negative, the above steps S404 through S408 are repeated.

The operator stops the panning of the camera at an appropriate position where an appropriate overlapping portion of the two adjacent images can be seen with the image of the peripheral boundary 27B on the screen 27A of the monitor 27, and turns ON the shutter switch 20 to store the current image. Every time the steps S404 through S408 are performed, the processor 30 compares the currently obtained displacement vector and the previously obtained displacement vector (stored in the internal register or the RAM 36)

so as to determine whether the operator stops the movement of the optical axis of the camera. If the difference between the two displacement vectors is larger than a threshold value, the result at the step **S410** is negative. If the difference between the two displacement vectors is less than the threshold value, the result at the step **S410** is affirmative.

The operator can easily carry out the divisional shooting process with the camera system by viewing both the current image and the peripheral boundary **27B** (or the preceding image) on the screen **27A** of the monitor **27**. A positional relation between the preceding image and the current image is clearly noticeable to the operator by viewing the peripheral boundary **27B** on the screen **27A** of the monitor **27** and the current image while the camera is panned in a desired direction. Therefore, the operator easily stops the movement of the optical axis of the camera at an appropriate position by viewing the image of the peripheral boundary **27B**, and turns ON the shutter switch **20** to store the current image.

When the result at the step **S410** is affirmative, the control is transferred to the step **S400**. The processor **30** at the step **S400** waits for the end of the image storage process at which the currently captured image is further stored in the image memory **28**.

The above-described embodiments of the present invention are applied to a digital camera. However, the present invention is not limited to the above-described embodiments. It is readily understood that the present invention is essentially applicable to a still-video camera and other camera systems which electronically store an image of an object and display the image on a display monitor. Further, variations and modifications of the above-described embodiments may be made without departing from the scope of the present invention.

The present invention is based on Japanese priority application No. 9-245522, filed on Sep. 10, 1997, the entire contents of which are hereby incorporated by reference.

What is claimed is:

1. A camera system comprising:

a display monitor for displaying an image of an object, taken by an optical unit and stored in a frame buffer, on a screen of the monitor;

a reading unit for reading a preceding image and a current image among a plurality of partially overlapping images, from the frame buffer, the preceding image and the current image containing a common element;

a determining unit for determining a positional relation between the preceding image and the current image based on a common pattern derived from the common element in the two adjacent images read by the reading unit; and

a displaying unit for displaying, when a shutter switch is turned on, a displaying unit image comprising the concurrent display of (i) a boundary image indicating a boundary of the preceding image on the screen of the monitor at a shifted position according to the positional relation determined by the determining unit, and (ii) the current image, wherein one of the plurality of partially overlapping images is defined by the overlap of the boundary image and the current image.

2. The camera system according to claim 1, wherein the determining unit performs a matching between corresponding blocks taken from an overlapping portion of the two adjacent images, so that a maximum-similarity common pattern in the two adjacent images is identified.

3. The camera system according to claim 1, wherein the determining unit performs a matching between corresponding blocks taken from an overlapping portion of the two adjacent images by checking intensities of individual pixels of the corresponding blocks.

4. The camera system according to claim 1, wherein the determining unit determines both coordinates of a central pixel of a maximum-similarity common pattern in the preceding image and coordinates of a central pixel of the maximum-similarity common pattern in the current image.

5. The camera system according to claim 1, further comprising:

a sensor for outputting an optical axis direction signal indicating a direction of an optical axis of the optical unit; and

a secondary determining unit for determining a positional relation between the preceding image and the current image based on a difference between the optical axis direction signal output by the sensor with respect to the current image and the optical axis direction signal output by the sensor with respect to the preceding image.

6. The camera system according to claim 1, wherein the determining unit determines a displacement vector, indicating a positional relation between the preceding image and the current image, based on a difference between coordinates of a central pixel of a maximum-similarity common pattern in the preceding image and coordinates of a central pixel of the maximum-similarity common pattern in the current image, and wherein the displaying unit displays an image of the displacement vector on the screen of the monitor with the current image concurrently displayed on the screen of the monitor.

7. The camera system according to claim 1, further comprising an image storing unit for storing an image of the object, taken by the optical unit, in an image memory, wherein the image storing unit stores auxiliary data, containing information indicating the positional relation from the determining unit, in the image memory, in addition to the image stored therein.

8. The camera system according to claim 5, further comprising an image storing unit for storing an image of the object, taken by the optical unit, in an image memory, wherein the image storing unit stores auxiliary data, containing information indicating the positional relation from the secondary determining unit, in the image memory, in additional to the image stored therein.

9. A divisional shooting method for a camera system in which at least two of partially overlapping images of an object, taken by an optical unit and stored in a frame buffer, are displayed, comprising the steps of:

reading a preceding image and a current image among the partially overlapping images, from the frame buffer, the preceding image and the current image containing a common element;

determining a positional relation between the preceding image and the current image based on a common pattern derived from the common element in the two adjacent images; and

displaying, when a shutter switch is turned on, a displaying unit image comprising the concurrent display of (i) a boundary image indicating a boundary of the preceding image on a screen of a display monitor at a shifted position according to the positional relation determined by the determining step, and (ii) the current image, wherein one of the at least two partially overlapping images is defined by the overlap of the boundary image and the current image.

10. The method according to claim 9, wherein, in the determining step, a matching between corresponding blocks taken from an overlapping portion of the two adjacent images is performed, so that a maximum-similarity common pattern in the two adjacent images is identified.

11. The method according to claim 9, wherein, in the determining step, a matching between corresponding blocks taken from an overlapping portion of the two adjacent images is performed by checking intensities of individual pixels of the corresponding blocks.

12. The method according to claim 9, wherein, in the determining step, both coordinates of a central pixel of a maximum-similarity common pattern in the preceding image and coordinates of a central pixel of the maximum-similarity common pattern in the current image are determined.

13. The method according to claim 9, further comprising the steps:

outputting an optical axis direction signal indicating a direction of an optical axis of the optical unit; and

determining a positional relation between the preceding image and the current image based on a difference between the optical axis direction signal output by the sensor with respect to the current image and the optical axis direction signal output by the sensor with respect to the preceding image.

14. The method according to claim 9, wherein, in the determining step, a displacement vector, indicating a positional relation between the preceding image and the current image, is determined based on a difference between coordinates of a central pixel of a maximum-similarity common pattern in the preceding image and coordinates of a central pixel of the maximum-similarity common pattern in the current image, and wherein, in the displaying step, an image of the displacement vector is displayed on the screen of the monitor with the current image concurrently displayed on the screen of the monitor.

15. The method according to claim 9, further comprising a step of storing an image of the object, taken by the optical unit, in an image memory, wherein auxiliary data, containing information indicating the positional relation from the determining unit, is stored in the image memory in addition to the image stored therein.

16. The method according to claim 13, further comprising a step of storing an image of the object, taken by the optical unit, in an image memory, wherein auxiliary data, containing information indicating the positional relation from the secondary determining unit, is stored in the image memory, in additional to the image stored therein.

17. The camera system according to claim 1, wherein the displaying unit displays a displacement vector, corresponding to the positional relation between the preceding image and the current image, during movement of an optical axis of the optical unit.

18. The method according to claim 9, further comprising the step of displaying a displacement vector, corresponding to the positional relation between the preceding image and the current image, during movement of an optical axis of the optical unit.

* * * * *

US006556705B1

(12) **United States Patent**
Shalom

(10) **Patent No.:** **US 6,556,705 B1**
(45) **Date of Patent:** **Apr. 29, 2003**

(54) **SYSTEM AND METHOD FOR ALIGNING A LOCALLY-RECONSTRUCTED THREE-DIMENSIONAL OBJECT TO A GLOBAL COORDINATE SYSTEM USING PARTIALLY-DETECTED CONTROL POINTS**

(75) Inventor: **Tamir Shalom**, Tel Aviv (IL)

(73) Assignee: **Cogni Tens, Ltd.**, Ramat Hasharon (IL)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/606,054**

(22) Filed: **Jun. 28, 2000**

Related U.S. Application Data

(60) Provisional application No. 60/142,347, filed on Jun. 28, 1999.

(51) **Int. Cl.**[7] **G06K 9/00**; G06T 15/00; G01C 3/14

(52) **U.S. Cl.** **382/154**; 382/294; 382/285; 345/419; 356/12

(58) **Field of Search** 382/294, 154, 382/284, 295, 285, 286, 293; 345/427, 486, 419, 672, 679; 356/12

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,531,520 A		7/1996	Grimson et al.	382/131
5,581,637 A		12/1996	Cass et al.	382/284
5,963,664 A	*	10/1999	Kumar et al.	382/154
6,061,468 A	*	5/2000	Kang	382/154
6,075,905 A	*	6/2000	Herman et al.	382/284
6,078,701 A	*	6/2000	Hus et al.	382/294
6,167,151 A	*	12/2000	Albeck et al.	382/154

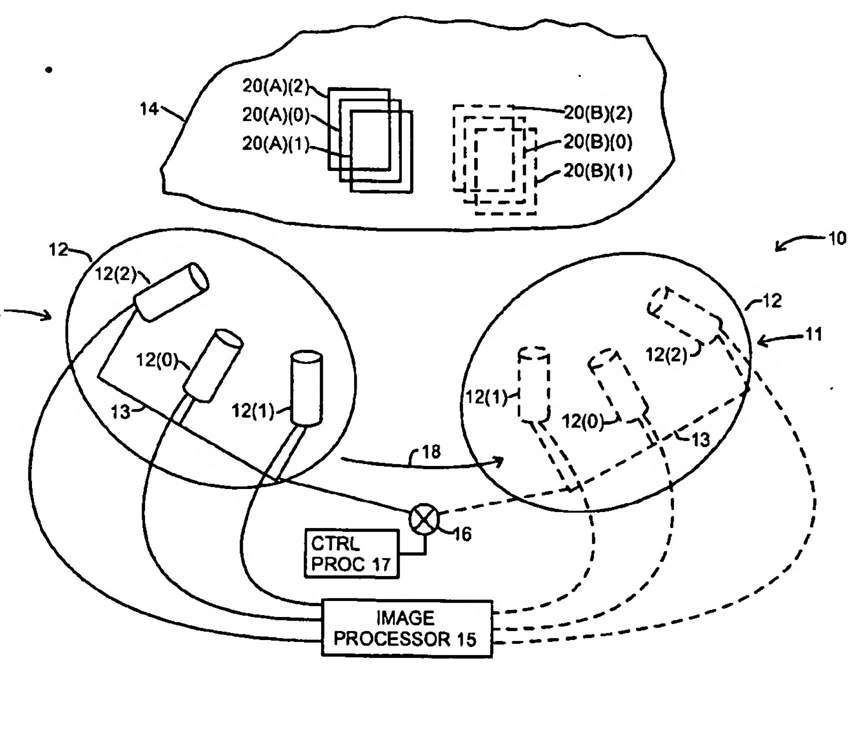
* cited by examiner

Primary Examiner—Bhavesh Mehta
Assistant Examiner—Yosef Kassa
(74) *Attorney, Agent, or Firm*—Richard A. Jordan

(57) **ABSTRACT**

A system and method are described for aligning a locally-reconstructed three-dimensional object, whose local reconstruction is relative to a local coordinate system, to a global coordinate system by using pre-mapped control points which are projected onto one or more of the images that may be used to generate the local reconstruction. A system includes a control point information generator and an alignment generator. The control point information generator is configured to identify in at least image associated with a local reconstruction of the object a projection of at least one control point in the scene onto the at least one image, and generate local projected coordinate information indicating coordinates of the projection in the at least one image. The alignment information generator is configured to utilize the local projected coordinate information and mapping information relating to global coordinate information indicating coordinates of the at least one control point relative to the global coordinate system to generate alignment information relating a local coordinate system associated with the local reconstruction to the global coordinate system.

24 Claims, 4 Drawing Sheets



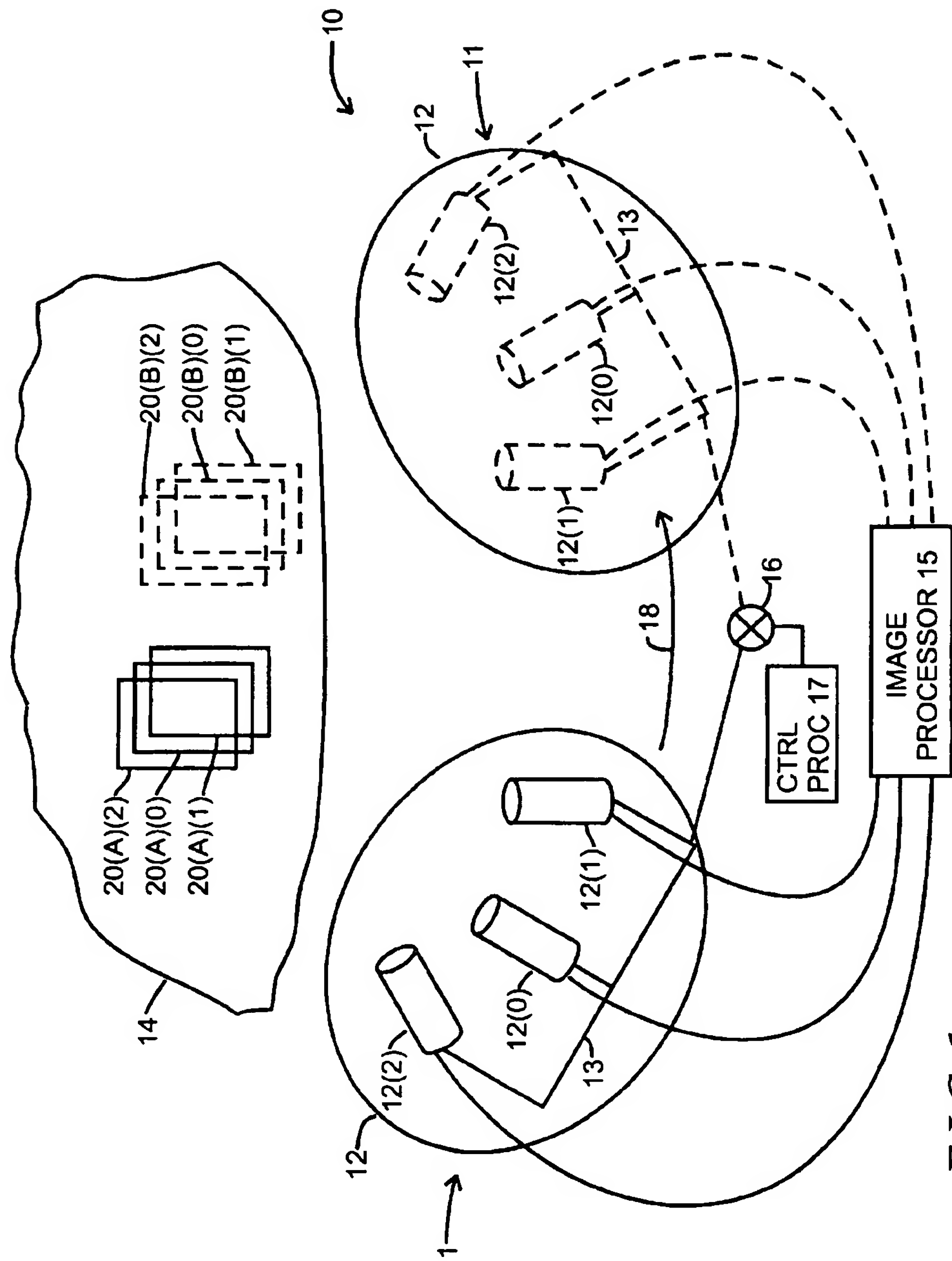

10
11
12
13
12(2)
12(0)
12(1)
16
18
CTRL PROC 17
IMAGE PROCESSOR 15
20(B)(2)
20(B)(0)
20(B)(1)
20(A)(2)
20(A)(0)
20(A)(1)
14

FIG. 1

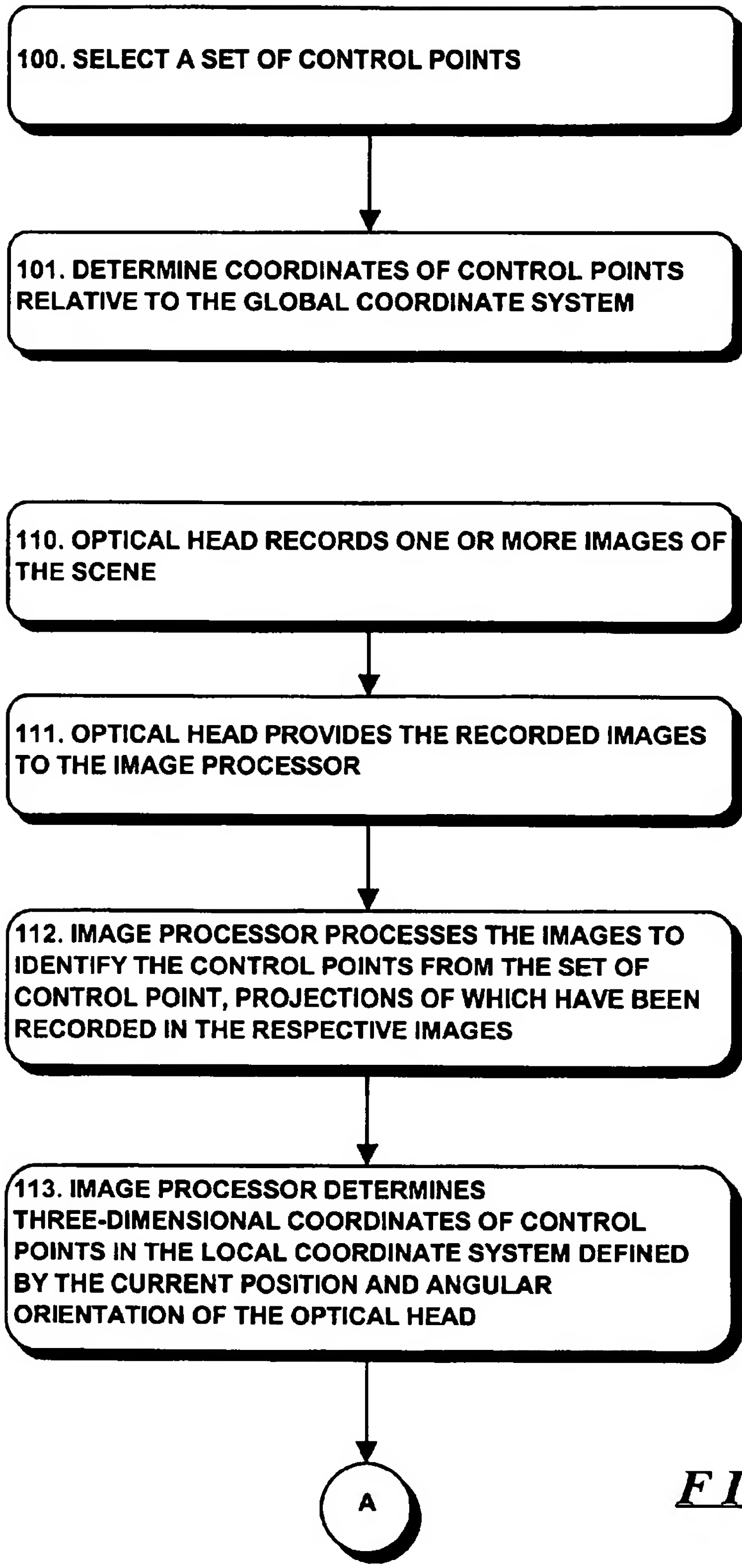

100. SELECT A SET OF CONTROL POINTS
101. DETERMINE COORDINATES OF CONTROL POINTS RELATIVE TO THE GLOBAL COORDINATE SYSTEM
110. OPTICAL HEAD RECORDS ONE OR MORE IMAGES OF THE SCENE
111. OPTICAL HEAD PROVIDES THE RECORDED IMAGES TO THE IMAGE PROCESSOR
112. IMAGE PROCESSOR PROCESSES THE IMAGES TO IDENTIFY THE CONTROL POINTS FROM THE SET OF CONTROL POINT, PROJECTIONS OF WHICH HAVE BEEN RECORDED IN THE RESPECTIVE IMAGES
113. IMAGE PROCESSOR DETERMINES THREE-DIMENSIONAL COORDINATES OF CONTROL POINTS IN THE LOCAL COORDINATE SYSTEM DEFINED BY THE CURRENT POSITION AND ANGULAR ORIENTATION OF THE OPTICAL HEAD
A

FIG. 2

FIG. 2A

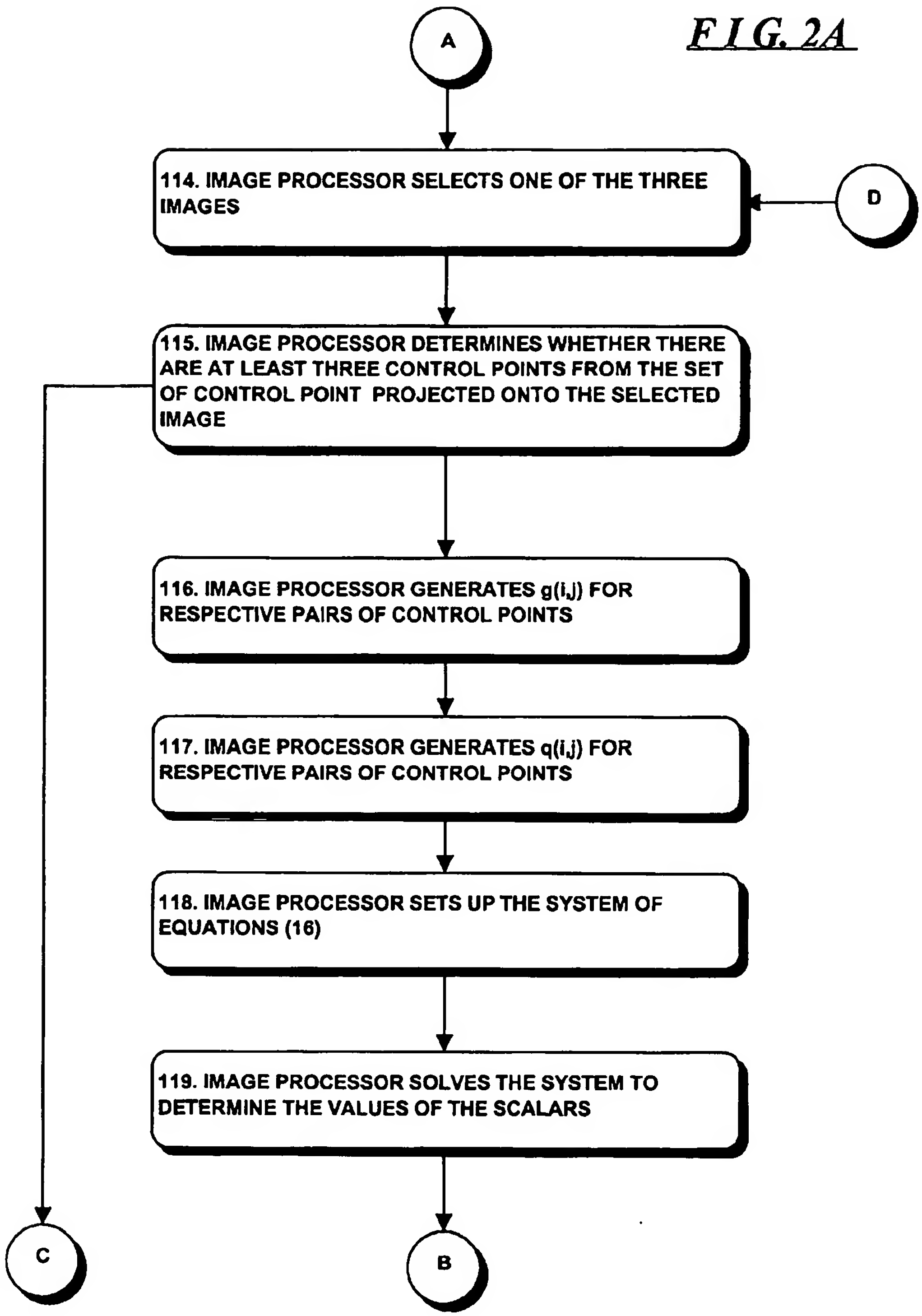
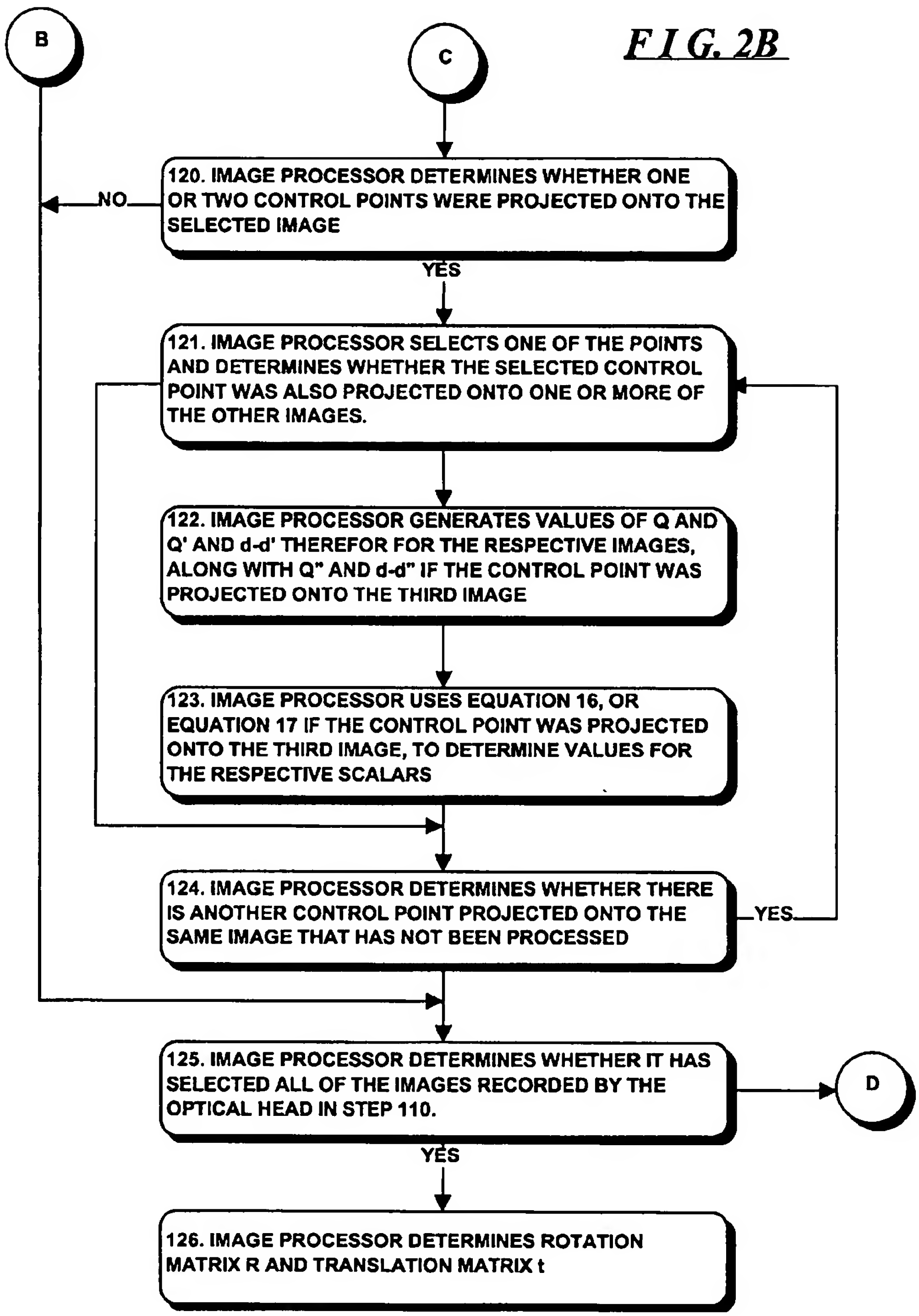

A
D
114. IMAGE PROCESSOR SELECTS ONE OF THE THREE IMAGES
115. IMAGE PROCESSOR DETERMINES WHETHER THERE ARE AT LEAST THREE CONTROL POINTS FROM THE SET OF CONTROL POINT PROJECTED ONTO THE SELECTED IMAGE
116. IMAGE PROCESSOR GENERATES g(i,j) FOR RESPECTIVE PAIRS OF CONTROL POINTS
117. IMAGE PROCESSOR GENERATES q(i,j) FOR RESPECTIVE PAIRS OF CONTROL POINTS
118. IMAGE PROCESSOR SETS UP THE SYSTEM OF EQUATIONS (16)
119. IMAGE PROCESSOR SOLVES THE SYSTEM TO DETERMINE THE VALUES OF THE SCALARS
C
B

FIG. 2B


B
C
120. IMAGE PROCESSOR DETERMINES WHETHER ONE OR TWO CONTROL POINTS WERE PROJECTED ONTO THE SELECTED IMAGE
NO
YES
121. IMAGE PROCESSOR SELECTS ONE OF THE POINTS AND DETERMINES WHETHER THE SELECTED CONTROL POINT WAS ALSO PROJECTED ONTO ONE OR MORE OF THE OTHER IMAGES.
122. IMAGE PROCESSOR GENERATES VALUES OF Q AND Q' AND d-d' THEREFOR FOR THE RESPECTIVE IMAGES, ALONG WITH Q" AND d-d" IF THE CONTROL POINT WAS PROJECTED ONTO THE THIRD IMAGE
123. IMAGE PROCESSOR USES EQUATION 16, OR EQUATION 17 IF THE CONTROL POINT WAS PROJECTED ONTO THE THIRD IMAGE, TO DETERMINE VALUES FOR THE RESPECTIVE SCALARS
124. IMAGE PROCESSOR DETERMINES WHETHER THERE IS ANOTHER CONTROL POINT PROJECTED ONTO THE SAME IMAGE THAT HAS NOT BEEN PROCESSED
YES
125. IMAGE PROCESSOR DETERMINES WHETHER IT HAS SELECTED ALL OF THE IMAGES RECORDED BY THE OPTICAL HEAD IN STEP 110.
D
YES
126. IMAGE PROCESSOR DETERMINES ROTATION MATRIX R AND TRANSLATION MATRIX t

SYSTEM AND METHOD FOR ALIGNING A LOCALLY-RECONSTRUCTED THREE-DIMENSIONAL OBJECT TO A GLOBAL COORDINATE SYSTEM USING PARTIALLY-DETECTED CONTROL POINTS

This application claims continuing data of No. 60/142,347, Jun. 28, 1999.

FIELD OF THE INVENTION

This invention relates generally to the field of reconstructing and/or manipulating surface geometries of one or more three dimensional objects in a scene, from a plurality of two dimensional images of the scene, and more particularly to a system and method for relating a local coordinate system associated with a reconstruction of at least a portion of the scene to a global coordinate system.

BACKGROUND OF THE INVENTION

Reconstruction and/or manipulation (generally, "reconstruction") of surface features of three-dimensional object(s) in a scene, from a plurality of two-dimensional images of the object(s), is useful in a number of applications. U.S. patent application Ser. No. 08/989,047, filed Dec. 11, 1997, in the names of Dan Albeck et al., and entitled "Apparatus And Method For 3-Dimensional Surface Geometry Reconstruction," now U.S. Pat. No. 5,617,151, issued Dec. 26, 2000, (hereinafter referred to as the "Albeck application") describes an apparatus for performing such reconstruction using a rig including an optical head of three cameras, using a tensor arrangement described in U.S. patent application Ser. No. 08/497,224, filed Jun. 30, 1995, in the name of Amnon Shashua, and entitled "Apparatus And Method For Recreating And Manipulating A 3D Object Based On A 2D Projection Thereof" now U.S. Pat. No. 5,821,943, issued Oct. 13, 1998, (hereinafter referred to as the "Shashua application") to generate information regarding reconstruction for the features of the object(s) from three images generated by the cameras. In the arrangement described in the Shashua application, the surface features that are reconstructed are defined by points that have coordinates in a coordinate system relative to one of the cameras in the rig. A problem arises in reconstruction if the surface features that are to be reconstructed cannot all be recorded by all of the cameras with the rig in one position. The apparatus described in the Albeck application provides a mechanism for moving the rig so as to allow the cameras to record sets of images of various portions of the surface(s) of the object(s). However, when the rig moves from one location, in which the cameras record a set of images of one portion of the object(s), to another location, in which the cameras record another set of images of another portion of the object(s), the coordinate system for the points defining the surface features of the various portions of the object(s) also changes.

In order to utilize the reconstruction information generated in the two "local" coordinate systems in a unitary manner in connection with the object(s) in the scene, it is necessary to relate the local coordinate systems to a common global coordinate system, which will allow all of the points of the various portions of the reconstructed object(s) to be related to the global coordinate system, effectively "stitching" the reconstructions together. The global coordinate system can conveniently be one of the two local coordinate systems that were used in the reconstruction, or it can be a third coordinate system, but in any case all of the points for

the various portions of the reconstructed object need to be related to the global coordinate system. When a rig, such as the rig described in the Albeck application is moved from one position to another, to facilitate recording of sets of images of different portions of the object(s), the movement comprises one or both of a translation and a rotation of the rig, both in three dimensions. If the translational and rotational movement of the rig can be controlled sufficiently precise, the relation of the coordinate system after movement to the coordinate system before movement can easily be determined. However, if, for example, the mass of the rig is sufficiently large, in a number of applications the movement of the rig cannot readily be controlled sufficiently precisely to allow sufficiently precise reconstruction.

U.S. patent application Ser. No. 09/165,687, filed Oct. 2, 1998, in the name of Tamir Shalom, et al., and entitled "System And Method For "Stitching" A Plurality Of Reconstructions Of Three-Dimensional Surface Features Of Object(s) In A Scene Defined Relative To Respective Coordinate Systems To Relate Them To A Common Coordinate System," now U.S. Pat. No. 6,201,541, issued Mar. 13, 2001, (hereinafter referred to as the Shalom application) describes an arrangement for relating a reconstruction of a portion of a scene to a global coordinate system. In the arrangement described in the Shalom application, a reconstruction of one portion of a scene is generated, which is related to the global coordinate system. Reconstructions of other portions can be related to the global coordinate system, but in the arrangement described in the Shalom application, the respective pairs of reconstructed portions will need to overlap with each other. That is, reconstructions successively displaced from the reconstruction which was originally related to the global coordinate systems will need to overlap, at least to some extent, with the overlapping portions providing information that is used to relate the local coordinate systems of the overlapping portions, thereby facilitating relation of the sequence of local coordinate systems to each other and, ultimately, to the global coordinate system. However, it will be appreciated that errors can develop in relating local coordinate systems of successive pairs of reconstructions, which can result in successively increasing errors in relating the local coordinate systems of the respective reconstructions to the global coordinate system. In addition, when the optical head is moved from one location to another to facilitate recording images from which the respective reconstructions are generated, the movement needs to be rigid, that is, the cameras need to maintain their respective orientations with respect to each other. If the optical head is not rigid, that is, if the cameras change their translational or rotational orientations with respect to each other, errors in relating the local coordinate systems of the respective pairs of reconstructions can also develop.

SUMMARY OF THE INVENTION

The invention provides a new and improved system and method for aligning a locally-reconstructed three-dimensional object to a global coordinate system using partially-detected control points.

In brief summary, the invention provides a system and method for aligning a locally-reconstructed three-dimensional object, whose local reconstruction is relative to a local coordinate system, to a global coordinate system by using pre-mapped control points which are projected onto one or more of the images that may be used to generate the local reconstruction. A system in accordance with the invention includes a control point information generator and an alignment generator. The control point information genera-

tor is configured to identify in at least image associated with a local reconstruction of said object a projection of at least one control point in the scene onto said at least one image, and generate local projected coordinate information indicating coordinates of said projection in said at least one image. The alignment information generator is configured to utilize said local projected coordinate information and mapping information relating to global coordinate information indicating coordinates of said at least one control point relative to said global coordinate system to generate alignment information relating a local coordinate system associated with said local reconstruction to said global coordinate system.

After the alignment information has been generated, the reconstruction in the local coordinate system can be aligned to the global coordinate system. This can be done with a plurality of local reconstructions, essentially stitching them together.

BRIEF DESCRIPTION OF THE DRAWINGS

This invention is pointed out with particularity in the appended claims. The above and further advantages of this invention may be better understood by referring to the following description taken in conjunction with the accompanying drawings, in which:

FIG. 1 schematically depicts a system for aligning a locally-reconstructed three-dimensional object to a global coordinate system using partially-detected control points, constructed in accordance with the invention;

FIG. 2 is a flow chart depicting operations performed by the system in connection with aligning a locally-reconstructed three-dimensional object to a global coordinate system using partially-detected control points.

DETAILED DESCRIPTION OF AN ILLUSTRATIVE EMBODIMENT

FIG. 1 is a diagram of a system **10** for aligning a locally-reconstructed three-dimensional object to a global coordinate system using partially-detected control points, constructed in accordance with the invention. With reference to FIG. 1, system **10** includes a rig **11** that supports an optical head **12** that comprises a plurality of cameras **12(0)** through **12(C)** (generally referred to by reference numeral **12**(*c*)) mounted on a common support **13**. The cameras **12**(*c*) are preferably focused on generally the same portion of a scene, including the same portion(s) of the surface(s) of one or more objects generally identified by reference numeral **14** in the scene, thereby to facilitate recording of a set of two-dimensional images of the respective portion(s) by the respective cameras **12**(*c*). The cameras **12**(*c*) provide the set of recorded images to an image processor **15** which generates, from the images provided thereto by the cameras **12**(*c*), information relating to a three-dimensional reconstruction of surface features of the respective object(s) **14** in the scene, for the respective portion(s) to which the cameras **12**(*c*) are focused. In one embodiment, the rig **11** and optical head are similar to the respective elements of the apparatus described in the aforementioned Albeck application. In that embodiment, the cameras **12**(*c*) preferably include CCD ("charge-coupled device") image recording devices, which provide image information in digital form to the image processor **15** for processing. In addition, in that embodiment, the image processor **15** includes suitably-programmed computing devices (such as a suitably-programmed general purpose computer) that generate the three-dimensional surface information from the set of two-dimensional images in a manner similar to that described in the aforementioned Shashua application.

The system **10** is further provided with a motor **16** which, under control of a control processor **17**, can move the rig **11** to facilitate direction of the cameras **12**(*c*) to another portion of the scene **14**, including respective other portion(s) of the surface(s) of respective object(s) in the scene **14** from which position the cameras **12**(*c*) can record a second set of images. The rig **11** before one such illustrative movement, indicated in FIG. 1 by the arrow associated with reference numeral **18**, is depicted in solid lines in FIG. 1, and rig **11** after the illustrative movement **18** depicted in dashed lines in FIG. 1. In addition, the set of images recorded by the respective cameras **12**(*c*) prior to the illustrative movement **18** of rig **11** is represented by images **20(A)(0)** through **20(A)(C)** (generally identified by reference numeral **20(A)**(*c*)), and the set of images recorded by the respective cameras **12**(*c*) after the illustrative movement **18** of rig **11** is represented by images **20(B)(0)** through **20(B)(C)** (generally identified by reference numeral **20(B)**(*c*)). The respective sets of images **20(A)**(*c*) and **20(B)**(*c*) are preferably not of coincident portions of the surface(s) of the object(s) **14** in the scene. Instead, the sets of images **20(A)**(*c*) and **20(B)**(*c*) will preferably be of different portions of the respective surface (s). The portions of the respective surface(s) represented by the sets of images **20(A)**(*c*) and **20(B)**(*c*) may, but need not, overlap.

The image processor **15** processes the images **20(A)**(*c*) provided by the cameras **12**(*c*) prior to the illustrative movement **18** to generate the three-dimensional surface information relating to the portion(s) of the object(s) depicted by the images **20(A)**(*c*). In addition, the image processor **15** processes the images **20(B)**(*c*) provided by the cameras **12**(*c*) after the illustrative movement **18** to generate the three-dimensional surface information relating to the portion(s) of the object(s) depicted by images **20(B)**(*c*). In one embodiment, operations performed by the image processor **15** in generating the three-dimensional surface information, in connection with the images recorded both before and after the illustrative movement **18**, correspond to those operations described in the aforementioned Shashua application, and the information so generated may represent a three-dimensional reconstruction of the portion(s) of the object(s) depicted in the respective set of images. In that embodiment, the image processor **15** generates, from a plurality of sets of points p_j, p_j' and p_j'' ("j" being an index) in images **20(A)(0)** through **20(A)(2)**, respectively, with points p_j, p_j' and p_j'' in each set corresponding to projections of a point P_j on a surface of an object in the scene **14** onto the respective images **20(A)(0)** through **20(A)(2)**, a tensor T_A. In generating the tensor T_A, the image processor **15** uses two-dimensional coordinates, in the respective images **20(A)**(*c*), of a selected number of sets of respective points p_j, p_j' and p_j''. As described in the Shashua application, seven sets of points p_j, p_j' and p_j'' will suffice to enable the image processor **15** to generate the tensor T_A. As also described in the Shashua application, using the tensor T_A and two-dimensional coordinates of any set of points p_j, p_j' and p_j'' in the images **20(A)**(*c*) which are projections of a point P_j (including, but not limited to, the seven sets of points used to generate the tensor T_A), the image processor **15** can determine the three-dimensional coordinates of the corresponding point P_j in the scene **14**. The three dimensional coordinates of point P_j will be in a coordinate system relative to one of the images **20(A)**(*c*), illustratively the image **20(A)(0)** recorded by camera **12(0)** before the illustrative movement **18**.

Similarly, the image processor generates, from coordinates a plurality of points q_j, q_j' and q_j'' in images 20(B)(0) through 20(B)(2), respectively, all corresponding to projections of respective points Q_j onto the respective images 20(B)(0) through 20(B)(2), a tensor T_B and coordinates for the respective points Q_j in a coordinate three-dimensional coordinate system relative to one of the images, illustratively image 20(B)(0), recorded by camera 12(0) after the illustrative movement 18. If optical head 12 is rigid under the movement, so that the cameras 12(*c*) will be in the same positions relative to each other before and after the illustrative movement 18, and if the three-dimensional coordinate systems are relative to images recorded by the same camera 12(0), then the tensor T_B will correspond to the tensor T_A. On the other hand, if the optical head 12 is not rigid under the illustrative movement 18, the tensor T_B will differ from the tensor T_A.

As noted above, the coordinates generated by the image processor 15 for points P_j generated using images 20(A)(*c*) are defined relative to a three-dimensional coordinate system associated with one of the cameras 12(*c*), for example, camera 12(0), in its position, prior to the illustrative movement 18, at which image 20(A)(0) was recorded, and coordinates generated by the image processor 15 for points Q_j from images 20(B)(*c*) are defined relative to a coordinate system that is preferably associated with the same camera 12(0), but which will be relative to the camera's position after the illustrative movement 18 at which image 20(B)(0) was recorded. In each case, the coordinates will be defined relative to two coordinate axes (illustratively referred to herein as "x" and "y" axes") which correspond to the plane of the respective image 20(A)(0) and 20(B)(0), and a third axis (illustratively referred to herein as a "z" axis) that is transverse thereto. It will be appreciated that the coordinate systems for the two sets of points P_j and Q_j in the scene 14 will differ. To be able to utilize both sets of points in a unitary fashion, it is desirable to associate them both to a common global coordinate system. The particular global coordinate system that is selected is not important. For simplicity and without loss of generality, the global coordinate system can be selected to be the local coordinate system before the illustrative movement 18 or after the illustrative movement 18. On the other hand, the global coordinate system may comprise a coordinate system separate from any of the local coordinate systems, illustratively, coordinate system 21.

Generally, the relation between the global coordinate system and the local coordinate systems consists of one or both of a translational component and/or a rotational component, both in three dimensional space. Thus, if, the coordinate system associated with camera 12(0), both before and after the illustrative movement 18, has an origin at the center of the image plane of the camera 12(0), the translational component comprises the position of the center of the image plane after the illustrative movement 18, in the coordinate system associated with the camera 12(0) before the illustrative movement 18. In addition, the rotational component corresponds to the angle of the image plane after the illustrative movement 18, relative to the image plane before the illustrative movement 18. It will be appreciated that, if the position of the origin after the illustrative movement 18 corresponds to the position of the origin before the illustrative movement 18, there is no translational component for the illustrative movement 18. Similarly, if the angular position of the coordinate axes after the illustrative movement 18 corresponds to the angular position of the coordinate axes before the illustrative movement 18, there is no rotational component for the illustrative movement 18.

It will be appreciated that, if the position and angular orientation of the rig 11 before and after the illustrative movement 18 can be controlled or determined (by, for example, mechanical or electronic sensors) to a sufficient degree of accuracy they (that is, the position and angular orientation) can be used to define the translational and rotational components to relate the coordinate system after the illustrative movement 18 to the coordinate system before the illustrative movement 18. However, in one embodiment, to provide a higher degree of accuracy, the image processor 15 for each reconstruction uses information relating to respective control points (not separately shown) that are distributed throughout the scene, images of at least some of which may be recorded one or more of the images 20(A)(*c*) and/or 20(B)(*c*) in each set. Generally, separately from the recording of the sets of images before and after the illustrative movement 18, during a mapping stage information is generated as to the positions of the control points relative to the global coordinate system. That information, along with information about the ones of the control points recorded on respective ones of images 20(A)(*c*) can be used to facilitate relation of the local coordinate system associated with the optical head 12 in the various positions to the global coordinate system. Operations performed by the image processor 15 in determining the translational and rotational components relating a local coordinate system to the global coordinate system will be described in connection with flowchart depicted in FIG. 2. Before proceeding to a description of FIG. 2 and by way of background, given a set of three-dimensional control points $\{G_i\}_{i=1}^n$ in the scene, coordinates in the global coordinate system for which are measured in the mapping stage. The three dimensional control points $\{G_i\}_{i=1}^n$ generally have respective projections p_i, p_i' and p_i'' in images 20(A)(*c*), although, as will be described below, not all control points need have projections in all three images. Assume further that the projection matrices for the respective images are represented by T, T' and T'', each being a three-by four matrix. Assume further that the local coordinate system differs from the global coordinate system by a rotation R and a translation t. In that case, during the reconstruction phase, the coordinates of the points p_i, p'_i and p''_i of the control points $\{G_i\}_{i=1}^n$ in the images 20(A)(*c*) are given by

$$T\cdot\begin{bmatrix} & R & & t \\ 0 & 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix} G_i \\ 1\end{bmatrix}=\lambda_i\begin{bmatrix} p_i \\ 1\end{bmatrix} \quad (1)$$

$$T'\cdot\begin{bmatrix} & R & & t \\ 0 & 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix} G_i \\ 1\end{bmatrix}=\lambda'_i\begin{bmatrix} p'_i \\ 1\end{bmatrix} \quad (2)$$

and

$$T''\cdot\begin{bmatrix} & R & & t \\ 0 & 0 & 0 & 1\end{bmatrix}\cdot\begin{bmatrix} G_i \\ 1\end{bmatrix}=\lambda''_i\begin{bmatrix} p''_i \\ 1\end{bmatrix} \quad (3)$$

respectively. It will be appreciated that the matrix

$$\begin{bmatrix} & R & & t \\ 0 & 0 & 0 & 1\end{bmatrix}$$

in equations (1) through (3) effectively operates to transform the coordinates of the control points $\{G_i\}_{i=1}^n$ from the global coordinate system to the local coordinate system following the transformation, and multiplying by the respective projection matrix T, T' and T'' provides the coordinates of the projections in the respective images 20(A)(*c*), up to a scale factor represented by the respective scalars λ_i, λ'_i and λ''_i.

If the projection matrices T, T' and T" are decomposed into respective three by three and three by one sub-matrices T=[S s], T'=[S' s'] and T"=[S" s"], where sub-matrices S, S' and S" are three-by-three matrices and sub-matrices s, s' and s" are one-by-three matrices (or vectors) equations (1) through (3) can be rewritten as

$$S\cdot(R\cdot G_i+t)+s=\lambda_i\begin{bmatrix}p_i\\1\end{bmatrix} \tag{4}$$

$$S''\cdot(R\cdot G_i+t)+s''=\lambda_i''\begin{bmatrix}p_i''\\1\end{bmatrix} \tag{5}$$

and

$$S'\cdot(R\cdot G_i+t)+s'=\lambda_i'\begin{bmatrix}p_i'\\1\end{bmatrix} \tag{6}$$

respectively. Since the matrices S, S', and S" are invertible, equations (4) through (6) can be re-written as

$$R\cdot G_i+t=\lambda_i S^{-1}\cdot\begin{bmatrix}p_i\\1\end{bmatrix}-S^{-1}\cdot s \tag{7}$$

$$R\cdot G_i+t=\lambda_i' S'^{-1}\cdot\begin{bmatrix}p_i'\\1\end{bmatrix}-S'^{-1}\cdot s' \tag{8}$$

and

$$R\cdot G_i+t=\lambda_i'' S''^{-1}\cdot\begin{bmatrix}p_i''\\1\end{bmatrix}-S''^{-1}\cdot s'' \tag{9}$$

respectively. Defining sets of three-dimensional points $\{P_i\}_{i=1}^n$, $\{P'_i\}_{i=1}^n$ and $\{P''_i\}_{i=1}^n$ by

$$P_i=S^{-1}\cdot\begin{bmatrix}p_i\\1\end{bmatrix},\ P_i'=S'^{-1}\begin{bmatrix}p_i'\\1\end{bmatrix}\text{ and }P_i''=S''^{-1}\begin{bmatrix}p_i''\\1\end{bmatrix}$$

and further defining $d=S^{-1}\cdot s$, $d'=S'^{-1}\cdot s'$ and $d''=S''^{-1}\cdot s''$, and noting that the left sides of equations (7) through (9) all equal $R\cdot G_i+t$, those equations (7) through (9) can be re-written

$$R\cdot G_i+t=\lambda_i P_i-d=\lambda'_i P'_i-d'=\lambda''_i P''_i-d''_i \tag{10}$$

Since the values of scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ are unknown, the points P_i, P'_i and P''_i can be normalized to

$$Q_i=\frac{P_i}{\|P_i\|},\ Q_i'=\frac{P_i'}{\|P_i'\|}\text{ and }Q_i''=\frac{P_i''}{\|P_i''\|},$$

where $\|P_i\|$, $\|P'_i\|$ and $\|P''_i\|$ represent the lengths of the vectors representing the points P_i, P'_i and P''_i, respectfully in the local coordinate system, and in that case equation 10 can be re-written as

$$R\cdot G_i+t=\lambda_i Q_i-d=\lambda'_i Q_i-d'=\lambda''_i-d'' \tag{11}$$

Using equation (10), the values of the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ and the components of the rotation matrix R and the translation matrix t can be obtained by minimizing the cost function

$$\sum_i \|(R\cdot G_i+t)-(\lambda_i Q_i-d)\|^2+ \tag{12}$$

-continued

$$\|(R\cdot G_i+t)-(\lambda_i'Q_i'-d')\|^2+\|(R\cdot G_i+t)-(\lambda_i''Q_i''-d'')\|^2.$$

Note that, if the rotation matrix R and translation matrix t are known, which may be the case if the position and orientation of the local coordinate system relative to the global coordinate system are known, the values of the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ can be determined by projecting the point $R\cdot G_i+t$ onto respective lines $\lambda_i Q_i-d$, $\lambda'_i Q'_i-d'$ and $\lambda''_i Q''_i-d''$. In that case, and since $Q^TQ=Q'^TQ'=Q''^TQ''=1$, where "T" represents the transpose operation,

$$\lambda_i=(R\cdot G_i+t+d)^T\cdot Q_i \tag{13}$$

$$\lambda'_i=(R\cdot G_i+t+d)^T\cdot Q'_i \tag{14}$$

and

$$\lambda''_i=(R\cdot G_i+t+d)^T\cdot Q''_i \tag{15}$$

On the other hand, if the values of the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ are known, the rotation matrix R and translation matrix t can be determined using the well-known Singular Value Decomposition (SVD) algorithm that fits the sets of points $\{\lambda_i Q_1-d\}_{i=1}^n$, $\{\lambda'_i Q'_i-d'\}_{i=1}^n$ and $\{\lambda''_i Q''_i-d''\}_{i=1}^n$ to the set of control points $\{G_i\}_{i=1}^n$, repeated three times for each of the three sets.

Accordingly, one can start with the values of the rotation matrix R and translation matrix t and use them to determine the values of the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$, or start with the values of the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ and use them to determine the values of the rotation matrix R and translation matrix t. In one embodiment, the image processor initially determines values of the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ and uses them to determine the values of the rotation matrix R and translation matrix t. Initially, it determines the values of the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ using the fact that the distance between two points in the scene is invariant when the optical head undergoes a rigid transformation (although the distance between the projections of the points in respective images may change) as follows. If projections of two different control points G_i and G_j appear in the same image, the distance between the points is given by $\|G_i-G_j\|=\|(R\cdot G_i+t)-(R\cdot G_j+t)\|$. Since $(R\cdot G_i+t)=(\lambda_i Q_i-d)$ and $(R\cdot G_j+t)=(\lambda_j Q_j-d)$, $\|G_i-G_j\|=\|\lambda_i Q_i-\lambda_j Q_j\|$.

On the other hand, if a control point G_i in the scene is projected onto two images, then the values of the scalars $\{\lambda_i\}_{i=1}^n$ and $\{\lambda'_i\}_{i=1}^n$ associated with the respective images can be determined using the fact that $\lambda_i Q_i-d=\lambda'_i Q'_i-d'$ for the two images. This provides three equations for the two unknowns λ_i and λ'_i, which can be solved in a conventional manner;

A third case is that each of two different control points G_i and G_j in the scene projects onto one of the two images. In that case, the distance between the two points is given by $\|G_i-G_j\|=\|(R\cdot G_i+t)-(R\cdot Q_j+t)\|$, in which case $\|G_i-G_j\|=\|(\lambda_i Q_i-d)-(\lambda'_j Q_j-d')\|$. However, since this case is less useful than the two described above, it will not be considered further.

Considering the first case, that is, the case in which projections of two different control points G_i and G_j appear in the same image, as noted above, the distance between the points is given by $\|G_i-G_j\|=\|\lambda_i Q_i-\lambda_j Q_j\|$. Defining $g_{i,j}=(G_i-G_j)^T(G_i-G_j)$ and $q_{i,j}=Q_i^T\cdot Q_j$, that equation can be re-written as

$$\lambda_i^2-2\lambda_i\lambda_j q_{i,j}+\lambda_j^2=g_{i,j} \tag{16}$$

In the case of "m" points, use of equation (16) provides

$$\begin{bmatrix} m \\ 2 \end{bmatrix}$$

(that is, the combination of "m" things taken "two" at a time) equations. Accordingly, if there are, for example, three points ("m"=3), there will be three equations. The appendix attached hereto includes the output of a program in the Maple symbolic language that shows one to obtain a fourth-degree equation in λ_i^2 that can be solved explicitly. If there are more than three points (that is, "m">3) there are more equations than unknowns and a methodology such as the Levenberg-Marquardt procedure can be used to obtain a solution for the system of equations defined by equation (16).

If the value of one of the scalars, illustratively λ_i, is already known, the values of the other scalars can be found by finding a point along a line that passes through the origin of the local coordinate system and has a direction towards point Q_j. In that case, the distance from the known point $\lambda_i Q_i$ is $\|G_i-G_j\|$. If the projection of $\lambda_i Q_i$ is further from that line than $\|G_i-G_j\|$, then the projection is the best solution, and, otherwise there are two equally good solutions.

Considering the second case, in which a point in the scene is projected onto two images, then

$$[Q_i - Q_i']\begin{bmatrix} \lambda_i \\ \lambda_i' \end{bmatrix} = [d - d']. \tag{17}$$

The matrix $[Q_i-Q'_i]$ in equation (17) is a three-row by two-column matrix (that is, Q_i and Q'_i are both three-element vectors) and the uniqueness and stability of the solution of the system defined by equation (17) depends on the rank of the matrix. Note that, if a point appears in all three images, the equation $\lambda_i Q_i-d=\lambda'_i Q'_i-d'=\lambda''_i Q_i-d''$ (reference equation (11) yields

$$\begin{bmatrix} Q_i & -Q_i' & 0 \\ Q_i & 0 & -Q_i'' \end{bmatrix}\begin{bmatrix} \lambda_i \\ \lambda_i' \\ \lambda''i \end{bmatrix} = \begin{bmatrix} d-d' \\ d-d'' \end{bmatrix}, \tag{18}$$

which is a system of six equations on three unknowns, a solution for which can be obtained in a conventional manner.

As noted above, after the values for the scalars $\{\lambda_i\}_{i=1}^n$, $\{\lambda'_i\}_{i=1}^n$ and $\{\lambda''_i\}_{i=1}^n$ have been determined, the rotation matrix R and translation matrix t can readily be determined using the well-known Singular Value Decomposition (SVD) algorithm that fits the three sets of points $\{\lambda_i Q_i-d\}_{i=1}^n$, $\{\lambda'_i Q'_i-d'\}_{i=1}^n$ and $\{\lambda''_i Q''_i-d''\}_{i=1}^n$ to the set of control points $\{G_i\}_{i=1}^n$, repeated three times once for each set.

With this background, operations performed by the system **10** will be described in connection with the flow chart in FIG. **2**. With reference to FIG. **2**, initially a series of operations are performed which together comprise the mapping stage. The mapping stage may be performed by the system **10** or by another device that can detect the control points and determine their coordinates in three dimensional space relative to the global coordinate system. In the mapping stage, a set of control points $\{G_i\}_{i=1}^n$, is initially selected (step **100**), and their coordinates relative to the global coordinate system determined (step **101**). Step **101** can be performed in a number of ways as will be apparent to those skilled in the art; several methodologies are described in U.S. Pat. No. 5,598,515, issued Jan. 28, 1997, in the name of Amnon Shashua, and entitled "System And Method For Reconstructing Surface Elements Of Solid Objects In A Three-Dimensional Scene From A Plurality Of Two Dimensional Images Of The Scene," and U.S. Pat. No. 5,821,943, issued Dec. 13, 1998, in the name of Amnon Shashua, and entitled "Apparatus And Method For Recreating And Manipulating A 3D Object Based On A 2D Projection Thereof, both assigned to the assignee of this application.

After the mapping stage has been completed, the system **10** can use the information determined for the control points during the mapping stage, and information determined for the control points as recorded in images **20(A)**(*c*) or **20(B)**(*c*), respectively, to determine the rotational matrix R and translation matrix t that relate the global coordinate system to the local coordinate system associated with respective images **20(A)**(*c*) or **20(B)**(*c*). In those operations, the optical head can record one or more images of the scene (step **110**) and provide the recorded images to the image processor (step **111**). The image processor, in turn, processes the images to identify the control points from the set of control point $\{G_i\}_{i=1}^n$, projections of which have been recorded in the respective images (step **112**) and determines their three-dimensional coordinates in the local coordinate system defined by the current position and angular orientation of the optical head (step **113**). The image processor then selects one of the three images (step **114**) and determines whether there are at least three control points from the set of control point $\{G_i\}_{i=1}^n$ projected onto the selected image (step **115**). If the image processor makes a positive determination in step **115**, it proceeds to process the image in accordance with the first case mentioned above. In that operation, the image processor generates $g_{i,j}=(G_i-G_j)^T(G_i-G_j)$ for the respective pairs of control points (step **116**) and $q_{i,j}=Q_i^T \cdot Q_j$ (step **117**), sets up the system of equations

$$\lambda_i^2-2\lambda_i\lambda_j q_{i,j}+\lambda_j^2=g_{i,j} \tag{19}$$

(step **118**), and solves the system to determine the values of the scalars λ_i and λ_j (step **119**). It will be appreciated that the methodology used by the image processor in solving the system of equations in step **119** will depend on whether three control points appear in the selected image or more than three control points. If three control points appear in the selected image, the image processor can use the methodology detailed in the attached appendix. On the other hand, if more than three control points appear in the selected image, the image processor makes use of the Levenberg-Marquardt methodology to solve the system of equations.

Returning to step **115**, if the image processor makes a negative determination in that step, it sequences to step **120** to determine whether one or two control points were projected onto the selected image. If the image processor makes a positive determination in step **120**, it sequences to step **121** in which it selects one of the points and determines whether the selected control point was also projected onto one or more of the other images. If the image processor makes a positive determination in step **121**, it generates values of Q_i and Q'_i and d−d' therefor for the respective images, along with Q''_i and d−d" if the control point was projected onto the third image (step **122**). Thereafter, the image processor uses equation 16, or equation 17 if the control point was projected onto the third image, to determine values for the respective scalars λ_i and λ'_i as well as λ''_i if the control point was projected onto the third image (step **123**).

After the image processor has performed steps **121** through **123** in connection with the control point that was selected in step **120** projected onto the image that had been

selected in step **114**, it determines whether there is another control point projected onto the same image that has not been processed (step **124**). If the image processor makes a positive determination in connection with step **124**, it returns to step **120** to select the other control point projected on the selected image. Thereafter, the image processor performs steps **121** through **123** in connection with the other control point projected onto the selected image.

Following

(i) step **119**;

(ii) step **124** if the image processor makes a negative determination in that step, or

(iii) step **120** if the image processor makes a negative determination in that step,

the image processor sequences to step **125** to determine whether it has selected all of the images recorded by the optical head in step **110**. If the image processor makes a negative determination in step **125**, it returns to step **114** to select the next image. The image processor will perform steps **115** through **125** in connection with each of the images recorded by the optical head in step **110**.

After the image processor has processed all of the images recorded in step **110**, it determines the rotation matrix R and translation matrix t using the well-known Singular Value Decomposition (SVD) algorithm that fits the sets of points $\{\lambda_i Q_i - d\}_{i=1}^{n}$, $\{\lambda'_i Q'_i - d'\}_{i=1}^{n}$ and $\{\lambda''_i Q''_i - d''\}_{i=1}^{n}$ to the set of control points $\{G_i\}_{i=1}^{n}$, repeated three times for each of the three sets of points (step **126**). The rotation matrix R and translation matrix t relate the local coordinate system to the global coordinate system and can be used to relate three-dimensional coordinates of other points in the scene in the local coordinate system, as determined by the image processor from the images recorded in step **110**, to the global coordinate system. This can be used, for example, to relate reconstructions of the scene made from a number of orientations, each of which may include a different portion of the scene, to the global coordinate system, effectively stitching them together.

The invention provides a number of advantages. In particular, the invention provides an arrangement for relating a local coordinate system to a global coordinate system, using control points information for which is obtained during a mapping stage. Since this can be accomplished for each position of the optical head **12**, there is no necessity of having reconstructions of overlapping portions of the scene to relate respective local coordinate systems to the global coordinate system. The relation between each local coordinate system to the global coordinate system is determined independently of any other local coordinate systems.

It will be appreciated that a number of modifications may be made to the system described above. For example, although the invention has been described in relation to the system depicted in FIG. 1 and described in the Albeck application, it will be appreciated that the invention can be used in connection with any arrangement which facilitates reconstruction of surface features of objects in a scene using images of the surface features. The invention can be used in connection with any arrangement that makes use of any number of images to facilitate the reconstruction,including arrangements that make use of three images, as in the arrangement described in the Shashua patent, arrangements that make use of two images, as well as arrangements that make use of different numbers of images.

In addition, it will be appreciated that the mapping information, which is determined during the mapping stage, may be developed in any manner and by any type arrangement. For example, the mapping information may be determined in whole or in part by the system **10** as described above. Alternatively, the mapping information may be developed by other arrangements that can determine the three dimensional coordinates of the control points from two dimensional images relative to the global coordinate system, or by other arrangements, such as directly measuring the positions of the control points relative to the global coordinate system.

It will be appreciated that a system in accordance with the invention can be constructed in whole or in part from special purpose hardware or a general purpose computer system, or any combination thereof, any portion of which may be controlled by a suitable program. Any program may in whole or in part comprise part of or be stored on the system in a conventional manner, or it may in whole or in part be provided in to the system over a network or other mechanism for transferring information in a conventional manner. In addition, it will be appreciated that the system may be operated and/or otherwise controlled by means of information provided by an operator using operator input elements (not shown) which may be connected directly to the system or which may transfer the information to the system over a network or other mechanism for transferring information in a conventional manner.

The foregoing description has been limited to a specific embodiment of this invention. It will be apparent, however, that various variations and modifications may be made to the invention, with the attainment of some or all of the advantages of the invention. It is the object of the appended claims to cover these and such other variations and modifications as come within the true spirit and scope of the invention.

APPENDIX

MAPLE program on 3 points fitting.

```
> restart;A12:=l1 ^ 2 - 2*l1*l2*q12 + l2 ^ 2 - g12;
            A12 := l1^2 - 2 l1 l2 q12 + l2^2 - g12
> A13:=l1 ^ 2 - 2*l1*l3*q13 + l3 ^ 2 - g13;
            A13 := l1^2 - 2 l1 l3 q13 + l3^2 - g13
> A23:=l2 ^ 2 - 2*l2*l3*q23 + l3 ^ 2 - g23;
            A23 := l2^2 - 2 l2 l3 q23 + l3^2 - g23
> A13-A23;
      l1^2 - 2 l1 l3 q13 - g13 - l2^2 + 2 l2 l3 q23 + g23
> l3:=solve(A13 - A23, l3);
```

$$l3 := 1/2\frac{l1^2 + g23 - g13 - l2^2}{-l2\ q23 + l1\ q13}$$

```
> B12:=collect(numer(simplify(A13)),l2);
B12 := l2^4 - 4 l1 q13 l2^3 q23 +
    (-2 g23 + 4 l1^2 - q13^2 + 4 l1^2 q23^2 - 4 g13 q23^2 - 2 l1^2 + 2 g13)
    l2^2 + (4 l1 q13 g23 q23 - 4 l1^3 q23 q13 + 4 l1 q13 g13 q23) l2 +
    2 l1^2 g23 + g13^2 4 l1^2 q13^2 g23 - 2 g23 g13 2 l1^2 g13 +
    l1^4 + g23^2
> B12elim4:=simplify(coeff(B12,l2,4)/coeff(A12,l2,2));
            B12elim4 := 1
> B12free4 := simplify(B12 - A12*l2 ^ 2*B12elim4);
B12free4 := - 4 l1 q13 l2^3 q23 - 2 g23 l2^2 + 4 l1^2 q13^2 l2^2 +
    4 l1^2 l2^2 q23^2 - 4 g13 l2^2 q23^2 - 3 l1^2 l2^2 + 2 g13 l2^2 +
    4 l1 q13 g23 l2 q23 - 4 l1^3 l2 q23 q13 + 4 l1 q13 g13 l2 q23 +
    2 l1^2 g23 + g13^2 - 4 l1^2 q13^2 g23 - 2 g23 g13 - 2 l1^2 g13 +
    l1^4 + g23^2 + 2 l2^2 l1 q12 + l2^2 g12
> B12elim3 := simplify(coeff(B12free4,l2,3)/coeff(A12,l2,2));
            B12elim3 := - 4 l1 q13 q23 + 2 l1 q12
> B12free3 := simplify(B12free4 - A12*l2*B12elim3);
B12free3 := 4 l1^2 l2^2 q23^2 + 2 l1^2 g23 - 2 l1^2 g13 3 l1^2 l2^2 + g23^2 -
    2 g23 - g13 - 2 g23 l2^2 + g13^2 + 2 g13 l2^2 + l2^2 g12 + l1^4 -
    4 l1^2 q13^2 g23 + 4 l1^2 q13^2 l2^2 4 g13 l2^2 q23^2 +
    4 l1 q13 g23 l2 q23 + 4 l1 q13 g13 l2 q23 + 4 l1^2 l2^2 q12^2 -
    2 l2 l1^3 q12 - 8 l1^2 l2^2 q12 q13 q23 - 4 l2 g12 l1^2 q13 q23 +
    2 l2 g12 l1 q12
> C12:=simplify(B12free3*coeff(A12,l2,2) -A12*coeff(B12free3,l2,2)) ;
```

APPENDIX-continued

MAPLE program on 3 points fitting.

```
C12 := - 4 l1^3 q13^2 + 4 l1^2 g23 - 4 l1^2 g13 + g23^2 - 2 g23 g13 + g13^2 -
    4 l1^2 q23^2 - 4 l1^2 g12 + 4 l1^4 - 4 l1^2 q13^2 g23 +
    4 l1 q13 g23 l2 q23 + 4 l1 q13 g13 l2 q23 - 8 l2 l1^3 q12 +
    4 l1^2 g13 q23^2 + 8 l1^3 l2 q12^3 + 4 g12 l1^2 q23^2 4 l1^4 q12^2 -
    4 l2 g12 l1 q13 q23 + 4 l2 g12 l1 q12 + 8 l1^4 q12 q13 q23 +
    8 l1^3 l2 q12 q23^2 - 2 g12 g23 + 2 g12 g13 + g12^2 +
    8 l1^3 l2 q12 q13^2 - 8 l1 l2 q12 g13 q23^2 -
    16 l1^3 l2 q12^2 q13 q23 - 4 l1 l2 q12 g23 + 4 l1 l2 q12 ql3 +
    4 g12 l1^2 q13^2 - 4 g12 g13 q23^2 + 4 g12 - l1^2 q12^2 -
    8 g12 l1^2 q12 q13 q23
> l2 := - coeff(C12,l2,0)/coeff(C12,l2,1);
l2 := - (-4 l1^4 q13^2 + 4 l1^2 g23 - 4 l1^2 g13 + g23^2 - 2 g23 g13 +
    g13^2 - 4 l1^4 q23^2 - 4 l1^2 g12 + 4 l1^4 4 l1^2 q13^2 g23 +
    4 l1^4 g13 q23^4 + 4 g12 l1^2 q23^2 - 4 l1^4 q12^2 +
    8 l1^4 q12 q13 q23 - 2 g12 g23 + 2 g12 g13 + g12^2 +
    4 g12 l1^2 q13^2 - 4 g12 g13 q23^2 + 4 g12 l1^2 q12^2 -
    8 g12 l1^2 q12 q13 q23)/(4 l1 q13 g23 q23 +
    4 l1 q13 g13 q23 - 8 l1^3 q12 + 8 l1^3 q12^3 - 4 g12 l1 q13 q23 +
    4 g12 l1 q12 + 8 l1^3 q12 q23^2 + 8 l1^3 q12 q13^2 -
    8 l1 q12 g13 q23^2 - 16 l1^3 q12^2 q13 q23 - 4 l1 q12 g23 +
    4 l1 q12 g13)
> A := simplify(numer(A12)):
> factor (coeff(A,l1,0);
(-4 g12 g13 q23^2 + g12^2 + 2 g12 g13 + g13^2 + g23^2 - 2 g12 g23 -
    2 g23 g13)^2
> coeff(A,l1,1);
                    0
> factor (coeff(A,l1,2)
-16 g12^3 q13^2 q23^2 - 24 g12^2 q13^2 g23 + 24 g12 q13^2 g23^2 +
    8 g12 q13^2 g13^2 + 16 g12^2 g13 q13^2 - 40 g12^2 g13 q12 q13 q23 -
    16 g12^2 q12^2 g13 q23^2 + 32 g12 q13 q23^2 q12 g13 g23 +
    32 g13^2 g12 q12 q23^2 q13 + 32 g12^2 g13 q12 q23^2 q13 -
    32 q12^2 g23 g13 g12 + 8 g13^2 q12 q13 q23 + 8 g12^2 q12 q13^2 q23
-
    8 g23^2 g12 q12 q13 q23 - 16 q12^2 g23^2 g13 q23^2 -
    8 g23 g13^2 q12 q13 q23 + 48 g23 g13 g12 q12 q13 q23 +
    8 g23^3 q12 q13 q23 - 8 g12^3 + 32 g12^2 q13^2 - g23 q23^2 -
    16 g12 q13^2 g23^2 q23^2 - 16 g12 q13^2 g13^2 q23^2 -
    32 g13^2 q23^4 g12 - 16 g12^2 q23^2 g23 - 32 g12^2 q23^2 g13 +
    56 g12^2 g13 q23^2 + 48 g12 g23 g13 + 56 g13^2 g12 q23^2 +
    8 g23^2 g13 q23^2 + 8 g23^2 g12 q23^2 + 16 g23^2 g13 q13^2 -
    16 923 g13^2 q23^2 - 8 g13^2 q13^2 g23 - 24 q12^2 g23 g13^2 . -
    64 g12 g23 g13 q23^2 - 24 g13^2 g12 + 16 q12^2 g23^2 g12 +
    8 g12^2 q13^2 + 24 g12^2 g13 + 8 g12^3 q23^2 - 24 g23^2 g13 -
    24 g23^2 g12 + 24 g23 g13^2 + 24 923 g12^2 - 8 g23^2 q13^2 +
    8 g13^3 q23^2 + 8 q12^2 g13^3 - 8 q12^2 g23^3 - 32 g12 g23 g13 q13^2 +
    32 q12^2 g23 g13^2 q23^2 - 40 g13^2 g12 q12 q13 q23 +
    24 q12^2 g23^2 g13 + 16 q12^2 g13^2 g12 - 16 q12^2 g13^3 q23^2 -
    8 g12^2 q12^2 g23 + 8 g12^2 q12^2 g13 - 8 g12^2 g23 q12 q13 q23 +
    8 g23^3 - 8 g13^3 - 8 q12 g23^2 q13 g13 q23
> coeff(A,l1,3)
                    0
> factor(coeff(A,l1,4));
64 q23^4 g12 g13 - 32 q12^3 g12 g13 q13 q23 - 32 g12 q12 q13^3 g13 q23 +
    64 q12^2 g12 q13^2 g13 q23^2 - 32 q12 g12^2 q13^3 q23 +
    16 g12^2 q12^2 q13^2 - 32 q12^2 g23^2 q13 q23 +
    64 q12^2 g12 q13^2 q23^2 g23^2 - 32 q13^4 g23 g12 + 48 q13^2 g23 g13 -
    48 g12 q13^2 g13 + 48 g23 g13 q23^2 + 48 g12 g23 q23^2 +
    48 g12^2 q13^2 q23^2 - 48 g12 q12^2 g13 - 64 q13^2 g23 q23^2 g12 -
    32 q13 g23 q23 g12 q12 - 32 q13 g23 q23^3 q12 g13 +
    16 q13 g23^2 q23 q12 - 32 q13 g23 q23 q12 g13 +
    16 q13^2 g23^2 q23^2 + 16 q13^2 g13^2 q23^2 + 32 g12 q12^2 g13 q23^2 +
    48 g12 q12^2 g23 + 16 g12^2 q13 q23 q12 - 8 g12^2 q12^2 -
    128 g12 q - l3 q23^2 q12 g13 + 16 q13 g13^2 q23 q12 +
    32 q13^2 g13 q23^2 g12 + 160 q13 g13 q23 g12 q12 -
    32 q13 g13^2 q23^3 q12 + 24 g23^2 - 48 g23 g13 + 24 g13^2 -
    40 q12^2 g13^2 - 32 q12^4 g23 g13 - 40 q12^2 g23^2 +
    80 g12 q13^2 g23 - 64 q12^2 g13 q23^2 g23 + 48 q12^2 g13^2 q23^2 +
    80 q12^2 g23g13 - 48 g12 g23 + 48 g12 g13 + 24 g12^2 -
    112 g12 g13 q23^2 - 40 g12^2 q13^2 + 16 g13^2 q23^4 - 40 g12^2 q23^2 +
    16 g12^2 q23^4 + 16 q13^4 g23^2 + 16 g12^2 q13^4 - 8 g13^2 q13^2 -
    40 q13^2 g23^2 - 40 g13^2 q23^2 - 8 g23^2 q23^2 + 16 q12^4 g23^2 +
    16 q12^4 g132 - 64 q122 g23 g13 q132 + 64 q12^3 g23 g13 q13 q23 -
    32 q12^3 g13^2 q13 q23 - 32 q12 q13^3 g23^2 q23 -
    32 q12 q13^3 g23 g13 q23 + 16 g12^2 q12^2 q23^2 +
    16 q12^2 g23^2 q23^2 + 48 q12^2 g23^2 q13^2 + 16 q12^2 g13^2 q13^2 -
    64 q12^2 q13^2 g23 g12 + 64 q12 q13^2 g23 g12 q23 +
```

APPENDIX-continued

MAPLE program on 3 points fitting.

```
    64 q12^2 g23 g13 q13^2 q23^2 - 32 g12 q12 q23^2 q13 g23 -
    32 q12 g12^2 q23^2 q13 - 32 q12^2 g12 g23 q13 q23
> coeff(A,l1,5)
                    0
> factor(coeff(A,l1,6)
-32 (-1 + q12^2 + q13^2 - 2 q12 q13 q23 + q23^2) (-q12^2 g23 +
    2 q12^2 q13^2 g23 + q12^2 g13 q13 g13 q23 q12 q13 g23 q23 q12 -
    g12 q12 q13 q23 - q13^2 g23 - g12 + q23^2 g12 - g13 + g23 +
    g13 q23^2 + g12 q13^2)
> coeff(A,l1,7)
                    0
> factor(coeff(A,l1,8)
        16 (-1 + q12^2 + q13^2 - 2 q12 q13 q23 + q23^2)^2
```

What is claimed as new and desired to be secured by Letters Patent of the United States is:

1. A system for aligning a locally-reconstructed three-dimensional object in a scene to a global coordinate system comprising:

A. a control point information generator configured to identify in at least image associated with a local reconstruction of said object a projection of at least one control point in the scene onto said at least one image, and generate local projected coordinate information indicating coordinates of said projection in said at least one image; and

B. an alignment information generator configured to utilize said local projected coordinate information and mapping information relating to global coordinate information indicating coordinates of said at least one control point relative to said global coordinate system to generate alignment information relating a local coordinate system associated with said local reconstruction to said global coordinate system.

2. A system as defined in claim 1 in which said alignment information includes rotation information indicating an angular orientation of said local coordinate system relative to said global coordinate system.

3. A system as defined in claim 1 in which said alignment information includes translation information indicating a translational displacement of said local coordinate system relative to said global coordinate system.

4. A system as defined in claim 1 in which the control point information generator is configured to generate the local projected coordinate information as the value of a plurality of scalars λ_i and λ_j relation to $\|G_i-G_j\|-\|\lambda_i Q_i-\lambda_j Q_j\|$, where $\|G_i-G_j\|$ represents a distance between control points G_i and G_j in the scene and

$$Q_i = \frac{P_i}{\|P_i\|},$$

where

$$P_i = S^{-1} \cdot \begin{bmatrix} p_i \\ 1 \end{bmatrix},$$

p_i representing a projection of control point G_i into the at least one image, and S represents a portion of a projection matrix defining projections from the scene into the at least one image.

5. A system as defined in claim 1 in which the control point information generator is configured to identify projections of one control point onto each of two images and

generate the control point information coordinate information therefrom.

6. A system as defined in claim 5 in which the control point information generator is configured to generate the local projected coordinate information as the value of a plurality of scalars λ_i and λ'_i in relation to $\lambda_i Q_i - d = \lambda'_i Q'_i - d'$, where

$$Q_i = \frac{P_i}{\|P_i\|}\left(P_i = S^{-1} \cdot \begin{bmatrix} p_i \\ 1 \end{bmatrix}\right)$$

and $d = S^{-1} \cdot s$ and $d' = S'^{-1} \cdot s'$, with "S" and "s" both representing portions of a respective projection matrix defining projections from the scene into one of the images, and "S'" and "s'" both representing portions of a respective projection matrix defining projections from the scene into the other of the images.

7. A system as defined in claim 5 in which the control point information generator is configured to identify projections of one control point onto each of three images and generate the control point information coordinate information therefrom.

8. A system as defined in claim 1 in which said alignment information generator is configured to generate said alignment information by using a Singular Value Decomposition (SVD) algorithm that fits sets of points $\{\lambda_i Q_i - d\}_{i=1}^n$, $\{\lambda'_i Q'_i - d'\}_{i=1}^n$ and $\{\lambda''_i Q''_i - d''\}_{i=1}^n$ to the control points $\{G_i\}_{i=1}^n$, where

$$Q_i = \frac{P_i}{\|P_i\|}\left(P_i = S^{-1} \cdot \begin{bmatrix} p_i \\ 1 \end{bmatrix}\right)$$

and $d = S^{-1} \cdot s$, $d' = S'^{-1} \cdot s'$ and $d'' = S''^{-1} \cdot s''$, with "S" and "s" both representing portions of a respective projection matrix defining projections from the scene into one of the images, and "S'" and "s'" both representing portions of a respective projection matrix defining projections from the scene into another of the images, and "S''" and "s''" both representing portions of a respective projection matrix defining projections from the scene into the third of the images.

9. A method for aligning a locally-reconstructed three-dimensional object in a scene to a global coordinate system comprising the steps of:

A. a control point information generating step in which is identified in at least image associated with a local reconstruction of said object a projection of at least one control point in the scene onto said at least one image, and is generated local projected coordinate information indicating coordinates of said projection in said at least one image; and

B. an alignment information generating step in which are utilized said local projected coordinate information and mapping information relating to global coordinate information indicating coordinates of said at least one control point relative to said global coordinate system to generate alignment information relating a local coordinate system associated with said local reconstruction to said global coordinate system.

10. A method as defined in claim 9 in which said alignment information includes rotation information indicating an angular orientation of said local coordinate system relative to said global coordinate system.

11. A method as defined in claim 9 in which said alignment information includes translation information indicating a translational displacement of said local coordinate system relative to said global coordinate system.

12. A method as defined in claim 9 in which the control point information generating step includes the step of generating the local projected coordinate information as the value of a plurality of scalars λ_i and λ_j in relation to $\|G_i - G_j\| = \|\lambda_i Q_i - \lambda_j Q_j\|$, where $\|G_i - G_j\|$ represents a distance between control points G_i and G_j in the scene and

$$Q_i = \frac{P_i}{\|P_i\|},$$

where

$$P_i = S^{-1} \cdot \begin{bmatrix} p_i \\ 1 \end{bmatrix},$$

p_i representing a projection of control point G_i into the at least one image, and S represents a portion of a projection matrix defining projections from the scene into the at least one image.

13. A method as defined in claim 9 in which the control point information generating step includes the steps of identifying projections of one control point onto each of two images and generating the control point information coordinate information therefrom.

14. A method as defined in claim 13 in which the control point information generating step includes the step of generating the local projected coordinate information as the value of a plurality of scalars λ_i and λ'_i in relation to $\lambda_i Q_i - d = \lambda'_i Q_i - d'$, where

$$Q_i = \frac{P_i}{\|P_i\|}\left(P_i = S^{-1} \cdot \begin{bmatrix} p_i \\ 1 \end{bmatrix}\right)$$

and $d = S^{-1} \cdot s$ and $d' = S'^{-1} \cdot s'$, with "S" and "s" both representing portions of a respective projection matrix defining projections from the scene into one of the images, and "S'" and "s'" both representing portions of a respective projection matrix defining projections from the scene into the other of the images.

15. A method as defined in claim 13 in which the control point information generating step includes the steps of identifying projections of one control point onto each of three images and generating the control point information coordinate information therefrom.

16. A method as defined in claim 9 in which said alignment information generating step includes the step of generating said alignment information by using a Singular Value Decomposition (SVD) algorithm that fits sets of points $\{\lambda_i Q_i - d\}_{i=1}^n$, $\{\lambda'_i Q'_i - d'\}_{i=1}^n$ and $\{\lambda''_i Q''_i - d''\}_{i=1}^n$ to the control points $\{G_i\}_{i=1}^n$, where

$$Q_i = \frac{P_i}{\|P_i\|}\left(P_i = S^{-1} \cdot \begin{bmatrix} p_i \\ 1 \end{bmatrix}\right)$$

and $d = S^{-1} \cdot s$, $d' = S'^{-1} \cdot s'$ and $d'' = S''^{-1} \cdot s''$, with "S" and "s" both representing portions of a respective projection matrix defining projections from the scene into one of the images, and "S'" and "s'" both representing portions of a respective projection matrix defining projections from the scene into another of the images, and "S''" and "s''" both representing portions of a respective projection matrix defining projections from the scene into the third of the images.

17. A computer program product for use in connection with a computer to provide a system for aligning a locally-reconstructed three-dimensional object in a scene to a global

coordinate system comprising a computer readable medium having encoded thereon:

A. a control point information generator module configured to enable said computer to identify in at least image associated with a local reconstruction of said object a projection of at least one control point in the scene onto said at least one image, and generate local projected coordinate information indicating coordinates of said projection in said at least one image; and

B. an alignment information generator module configured to enable the computer to utilize said local projected coordinate information and mapping information relating to global coordinate information indicating coordinates of said at least one control point relative to said global coordinate system to generate alignment information relating a local coordinate system associated with said local reconstruction to said global coordinate system.

18. A system as defined in claim **17** in which said alignment information includes rotation information indicating an angular orientation of said local coordinate system relative to said global coordinate system.

19. A system as defined in claim **17** in which said alignment information includes translation information indicating a translational displacement of said local coordinate system relative to said global coordinate system.

20. A system as defined in claim **17** in which the control point information generator module is configured to enable said computer to generate the local projected coordinate information as the value of a plurality of scalars λ_i and λ_j in relation to $\|G_i-G_j\|=\|\lambda_iQ_i-\lambda_jQ_j\|$, where $\|G_i-G_j\|$ represents a distance between control points G_i and G_j in the scene and

$$Q_i = \frac{P_i}{\|P_i\|},$$

where

$$P_i = S^{-1}\cdot\begin{bmatrix} p_i \\ 1 \end{bmatrix},$$

p_i representing a projection of control point G_i into the at least one image, and S represents a portion of a projection matrix defining projections from the scene into the at least one image.

21. A system as defined in claim **17** in which the control point information generator module is configured to enable said computer to identify projections of one control point onto each of two images and generate the control point information coordinate information therefrom.

22. A system as defined in claim **21** in which the control point information generator module is configured to enable said computer to generate the local projected coordinate information as the value of a plurality of scalars λ_i and λ'_i in relation to $\lambda_iQ_i-d=\lambda'_iQ'_i-d'$, where

$$Q_i = \frac{P_i}{\|P_i\|}\left(P_i = S^{-1}\cdot\begin{bmatrix} p_i \\ 1 \end{bmatrix}\right)$$

and $d=S^{-1}\cdot s$ and $d'=S'^{-1}\cdot s'$, with "S" and "s" both representing portions of a respective projection matrix defining projections from the scene into one of the images, and "S'" and "s'" both representing portions of a respective projection matrix defining projections from the scene into the other of the images.

23. A system as defined in claim **21** in which the control point information generator module is configured to enable said computer to identify projections of one control point onto each of three images and generate the control point information coordinate information therefrom.

24. A system as defined in claim **17** in which said alignment information generator module is configured to enable said computer to generate said alignment information by using a Singular Value Decomposition (SVD) algorithm that fits sets of points $\{\lambda_iQ_i-d\}_{i=1}^n$, $\{\lambda'_iQ'_i-d'\}_{i=1}^n$ and $\{\lambda''_iQ''_i-d''\}_{i=1}^n$ to the control points $\{G_i\}_{i=1}^n$, where

$$Q_i = \frac{P_i}{\|P_i\|}\left(P_i = S^{-1}\cdot\begin{bmatrix} p_i \\ 1 \end{bmatrix}\right)$$

and $d=S^{-1}\cdot s$, $d'=S'^{-1}\cdot s'$ and $d''=S''^{-1}\cdot s''$, with "S" and "s" both representing portions of a respective projection matrix defining projections from the scene into one of the images, and "S'" and "s'" both representing portions of a respective projection matrix defining projections from the scene into another of the images, and "S''" and "s''" both representing portions of a respective projection matrix defining projections from the scene into the third of the images.

* * * * *

US006016147A

United States Patent [19]

Gantt

[11] **Patent Number: 6,016,147**

[45] **Date of Patent: *Jan. 18, 2000**

[54] **METHOD AND SYSTEM FOR INTERACTIVELY DETERMINING AND DISPLAYING GEOMETRIC RELATIONSHIPS BETWEEN THREE DIMENSIONAL OBJECTS BASED ON PREDETERMINED GEOMETRIC CONSTRAINTS AND POSITION OF AN INPUT DEVICE**

[75] Inventor: **Brian D. Gantt**, Travis County, Tex.

[73] Assignee: **Autodesk, Inc.**, San Rafael, Calif.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/744,241**

[22] Filed: **Nov. 5, 1996**

Related U.S. Application Data

[63] Continuation-in-part of application No. 08/436,158, May 8, 1995, Pat. No. 5,572,639.

[51] **Int. Cl.**[7] **G06T 15/70**

[52] **U.S. Cl.** **345/420**; 345/419

[58] **Field of Search** 345/145, 483, 345/421, 422, 435

[56] **References Cited**

U.S. PATENT DOCUMENTS

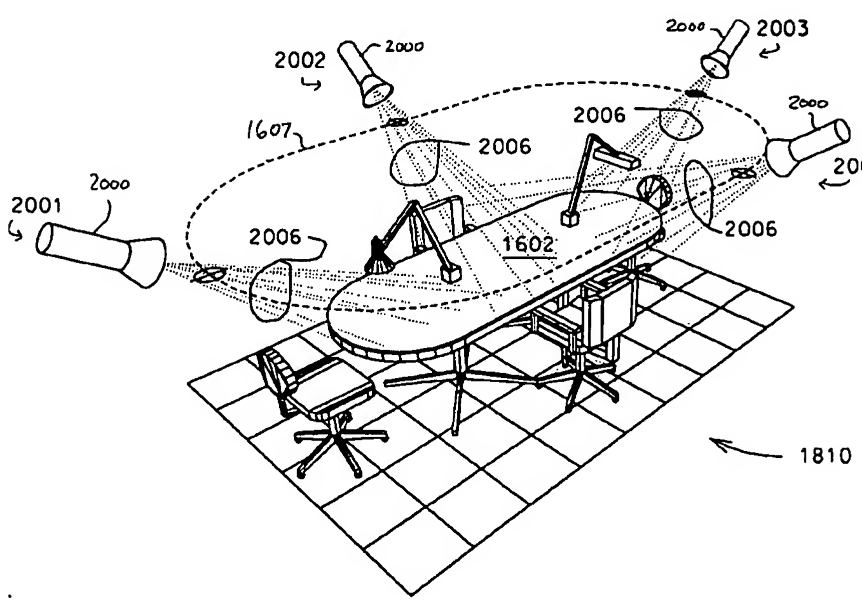
4,951,231	8/1990	Dickinson et al. .	
5,101,436	3/1992	DeAguiar et al. .	
5,123,087	6/1992	Newell et al. .	
5,133,052	7/1992	Bier et al. .	
5,237,647	8/1993	Roberts et al. .	
5,325,475	6/1994	Poggio et al. .	
5,371,845	12/1994	Newell et al. .	
5,396,590	3/1995	Kreegar .	
5,513,303	4/1996	Robertson et al.	345/419
5,712,965	1/1998	Fujita et al.	345/419
5,734,805	3/1998	Isensee et al.	345/433
5,734,806	3/1998	Naryanaswami	345/421
5,734,807	3/1998	Sumi	345/433

Primary Examiner—Mark R. Powell
Assistant Examiner—Huedung X. Cao
Attorney, Agent, or Firm—Gates & Cooper

[57] **ABSTRACT**

A system and method of interactively determining and displaying geometric relationships between three dimensional (3D) objects includes the steps of and apparatus for detecting the position of an input device, moving a selected 3D graphic object relative to a graphic pointing symbol in a 3D representation based on position of the input device, determining if the selected graphic object is moved to occlude an underlying 3D graphic object, and positioning and displaying the selected graphic object with respect to the underlying graphic object according to predetermined geometric constraints and the position of the input device. The system and method further dynamically moves and displays the selected graphic object according to movement of the input device and the predetermined geometric constraints while the selected graphic object occludes the underlying graphic object. The selected graphic object clings to the underlying graphic object, and is moved about the underlying graphic object corresponding to movement of the input device. The selected object may be a logical object, such as a logical camera or light source. For example, a camera object is placed based on a geometric element, and the display is changed to the viewpoint of the camera. The operator then interactively changes the display simply by moving the input device, where the display is automatically updated based on movement of the camera object.

16 Claims, 24 Drawing Sheets



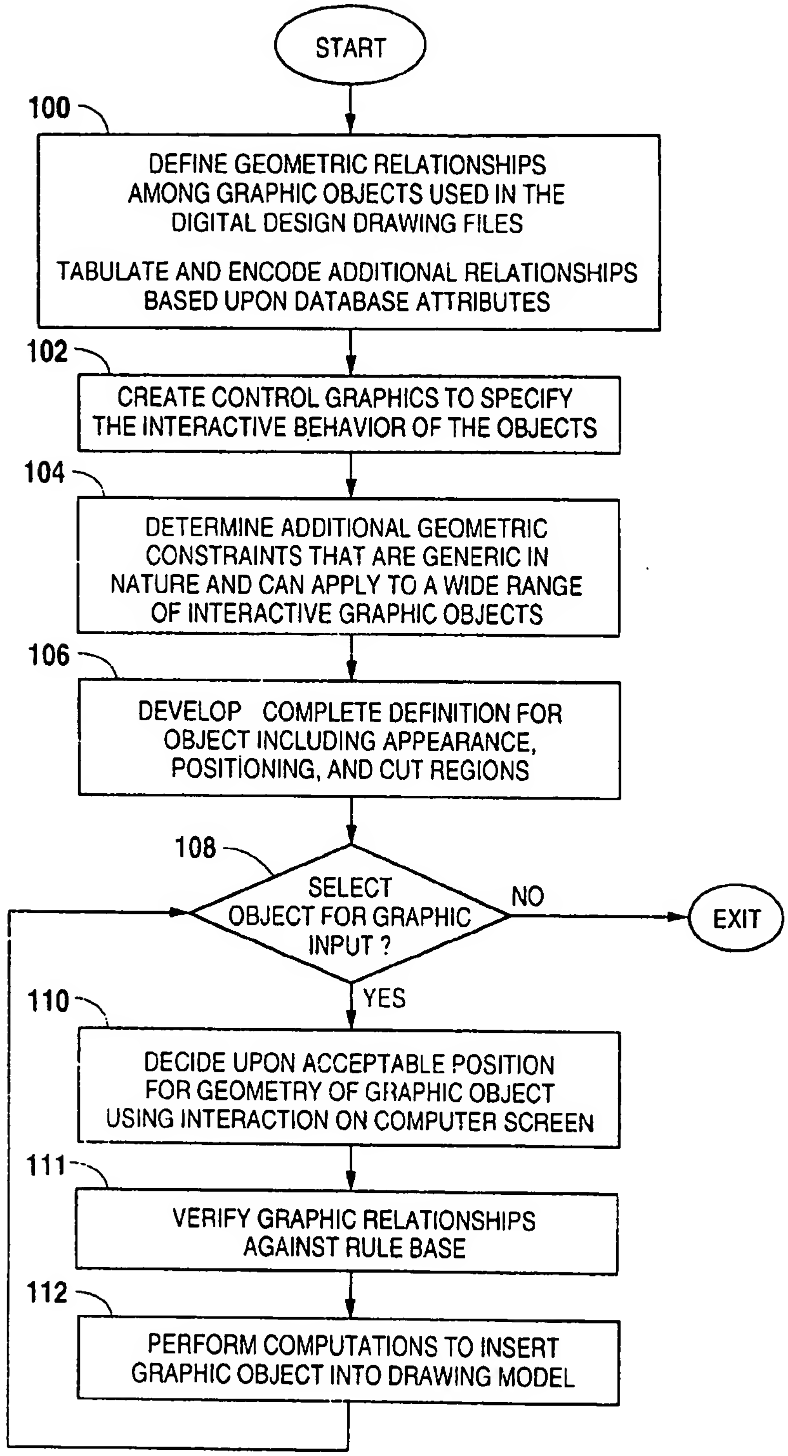

START
100
DEFINE GEOMETRIC RELATIONSHIPS AMONG GRAPHIC OBJECTS USED IN THE DIGITAL DESIGN DRAWING FILES
TABULATE AND ENCODE ADDITIONAL RELATIONSHIPS BASED UPON DATABASE ATTRIBUTES
102
CREATE CONTROL GRAPHICS TO SPECIFY THE INTERACTIVE BEHAVIOR OF THE OBJECTS
104
DETERMINE ADDITIONAL GEOMETRIC CONSTRAINTS THAT ARE GENERIC IN NATURE AND CAN APPLY TO A WIDE RANGE OF INTERACTIVE GRAPHIC OBJECTS
106
DEVELOP COMPLETE DEFINITION FOR OBJECT INCLUDING APPEARANCE, POSITIONING, AND CUT REGIONS
108
SELECT OBJECT FOR GRAPHIC INPUT ?
NO
EXIT
YES
110
DECIDE UPON ACCEPTABLE POSITION FOR GEOMETRY OF GRAPHIC OBJECT USING INTERACTION ON COMPUTER SCREEN
111
VERIFY GRAPHIC RELATIONSHIPS AGAINST RULE BASE
112
PERFORM COMPUTATIONS TO INSERT GRAPHIC OBJECT INTO DRAWING MODEL

Fig. 1

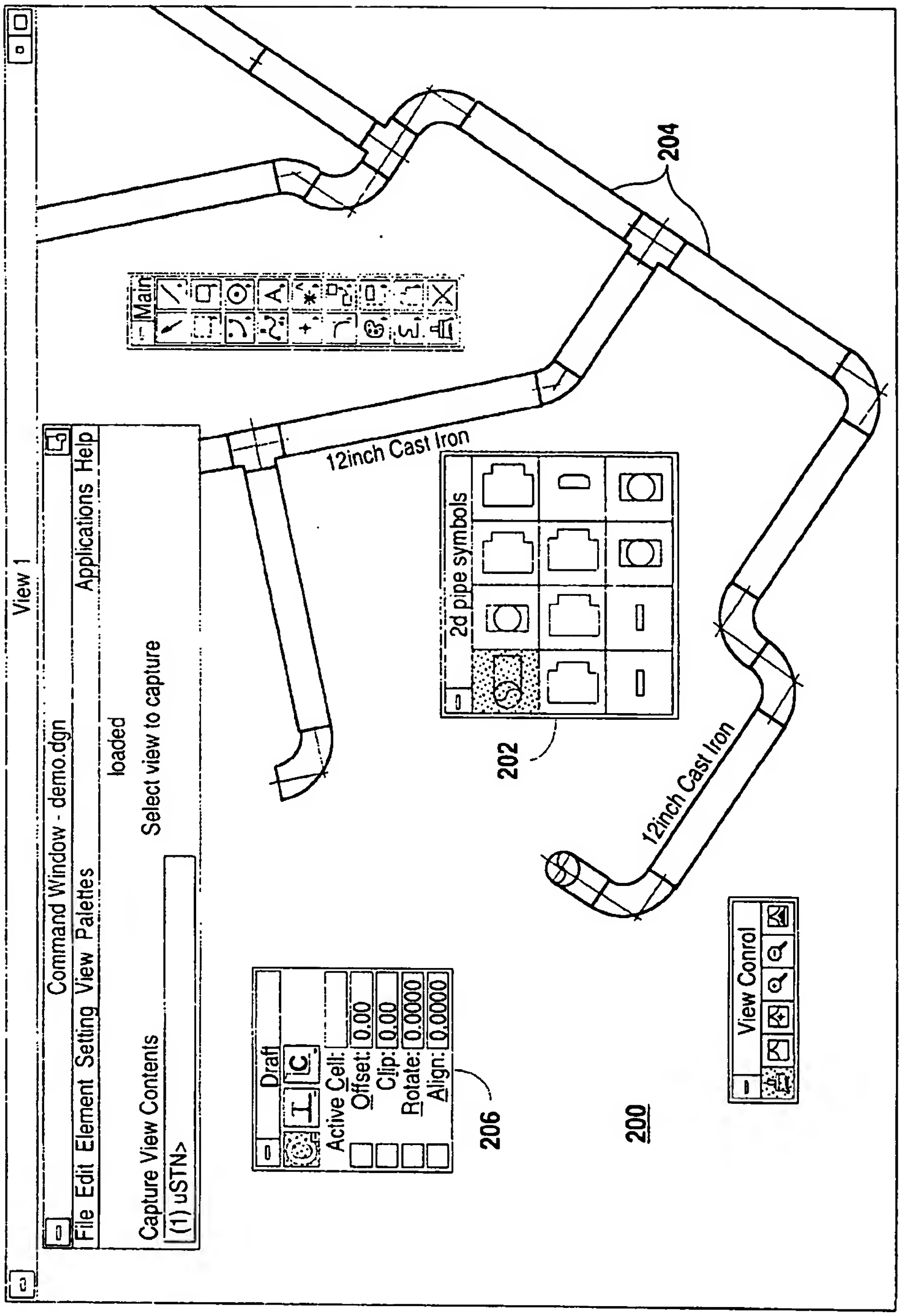

View 1
Command Window - demo.dgn
File Edit Element Setting View Palettes Applications Help
loaded
Select view to capture
Capture View Contents
(1) uSTN>
Main
12inch Cast Iron
2d pipe symbols
12inch Cast Iron
204
202
Draft
Active Cell:
Offset: 0.00
Clip: 0.00
Rotate: 0.0000
Align: 0.0000
206
200
View Conrol

Fig. 2

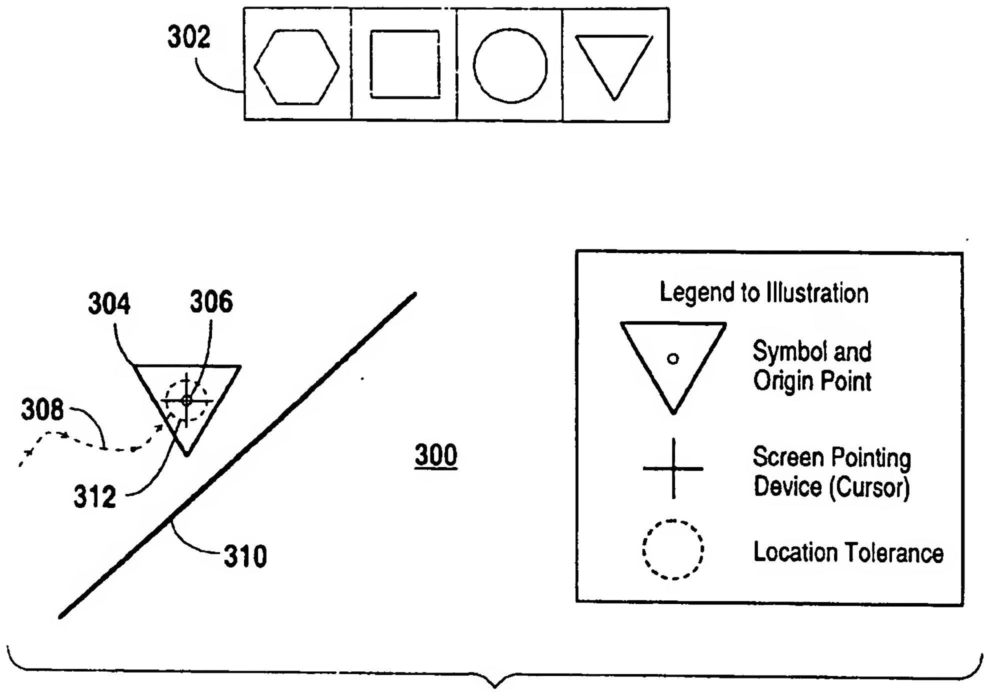

302
304
306
308
312
310
300
Legend to Illustration
Symbol and Origin Point
Screen Pointing Device (Cursor)
Location Tolerance

Fig. 3A

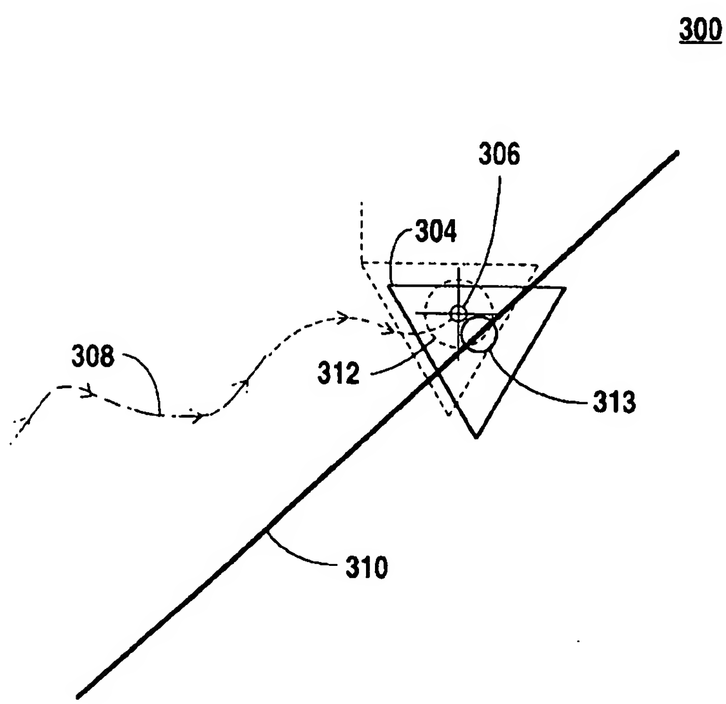

300
306
304
308
312
313
310

Fig. 3B

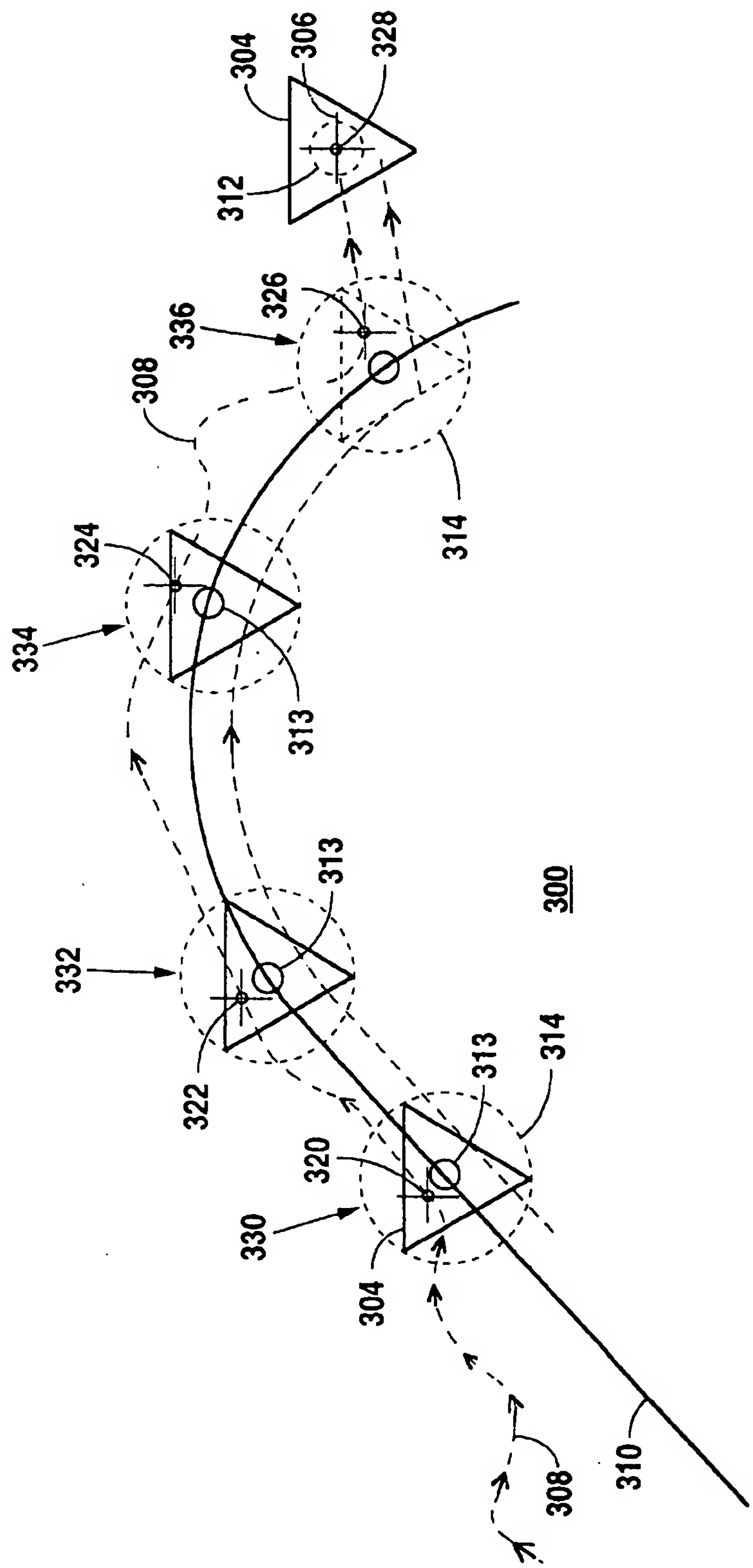

304
306
328
312
326
336
308
324
334
313
314
300
332
313
322
313
314
320
330
304
308
310

Fig. 3C

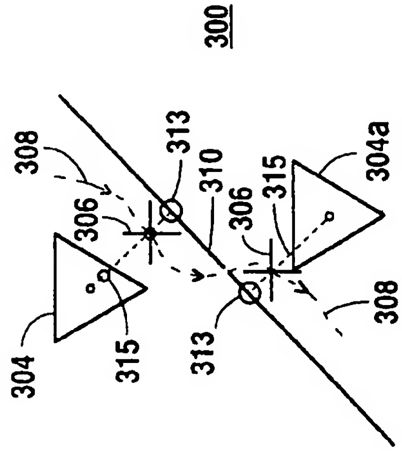


Fig. 3E

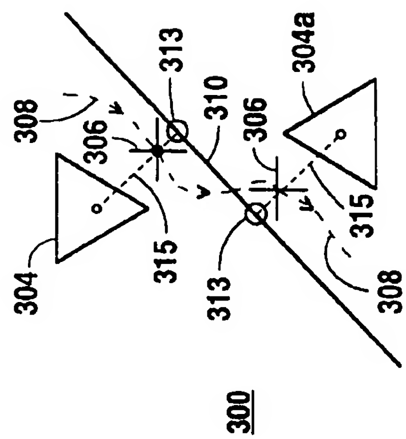


Fig. 3F

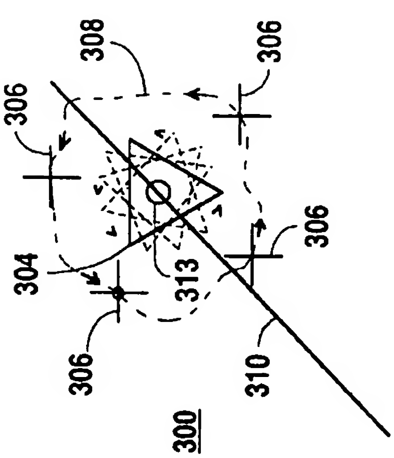


Fig. 3D

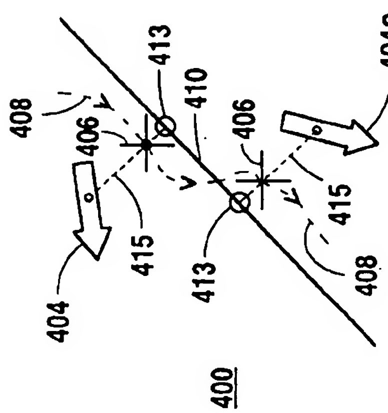


Fig. 4A

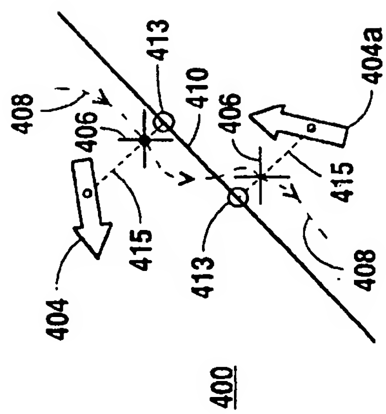


Fig. 4B

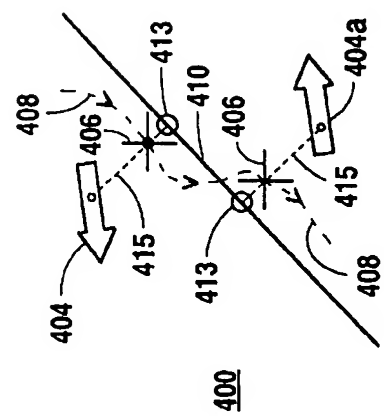


Fig. 4C

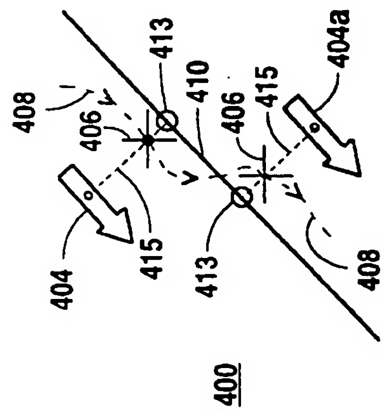


Fig. 4D

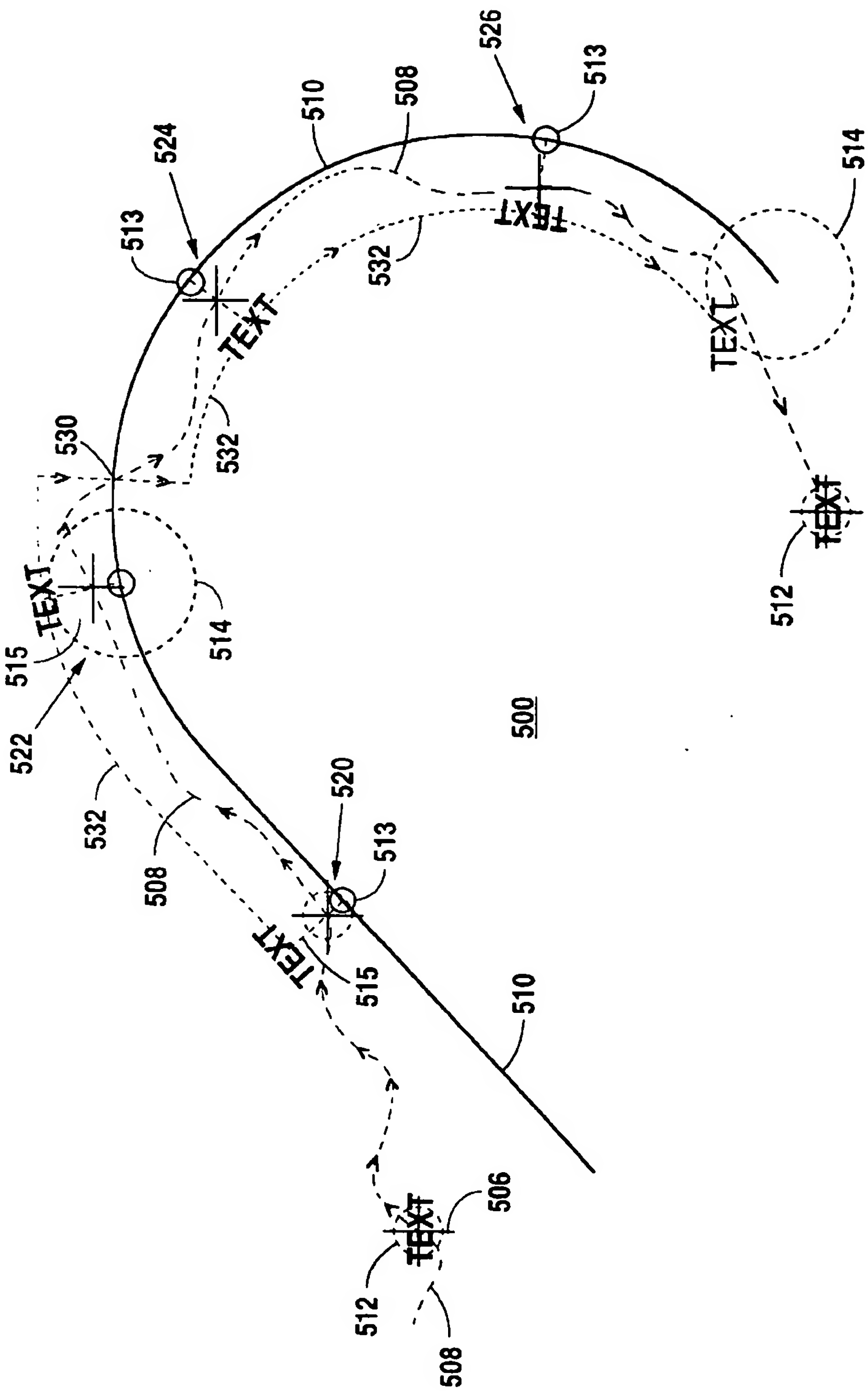


Fig. 5

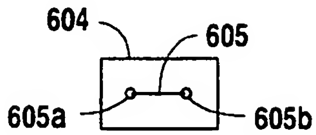


Fig. 6A

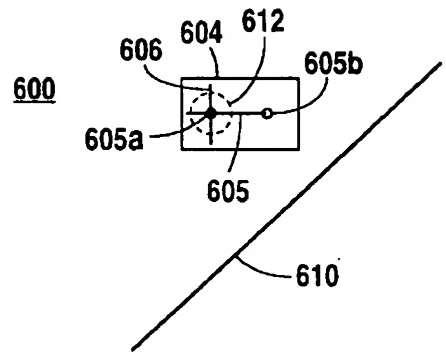


Fig. 6B

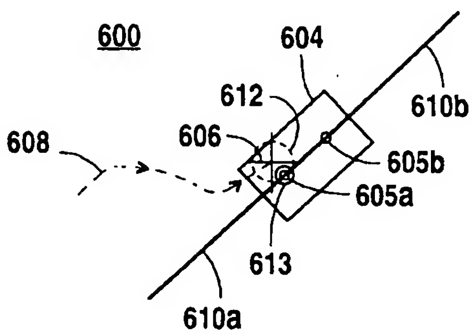


Fig. 6C

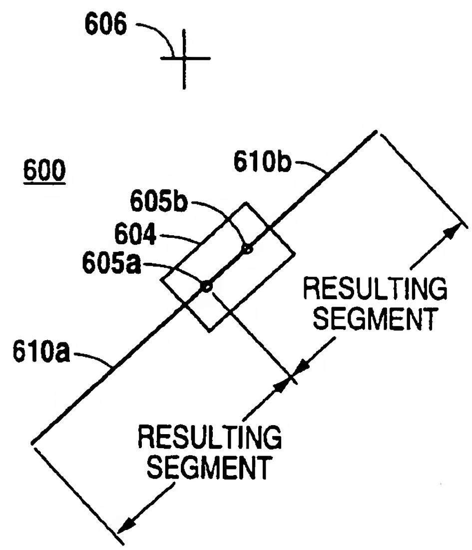


Fig. 6D

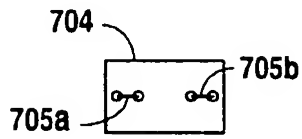


Fig. 7A

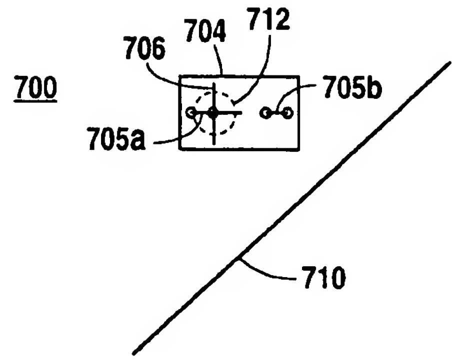


Fig. 7B

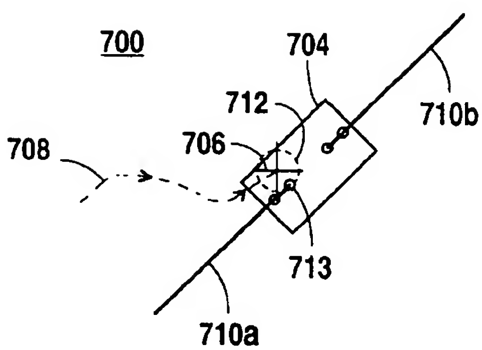


Fig. 7C

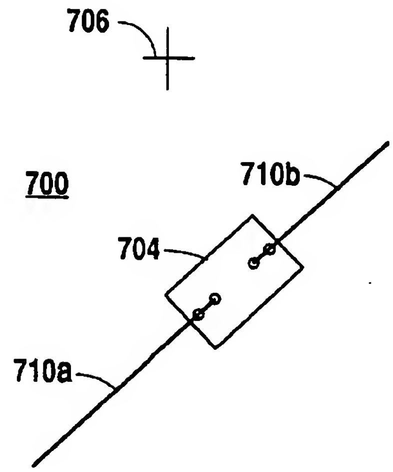


Fig. 7D

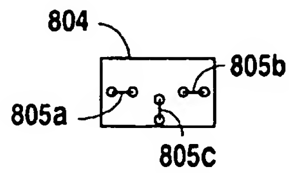


Fig. 8A

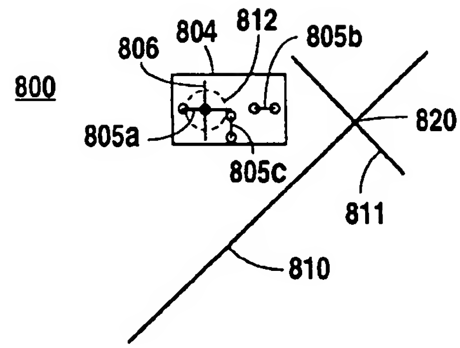


Fig. 8B

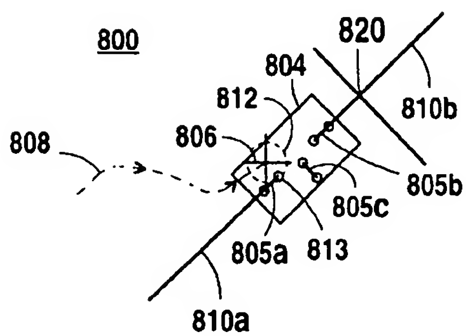


Fig. 8C

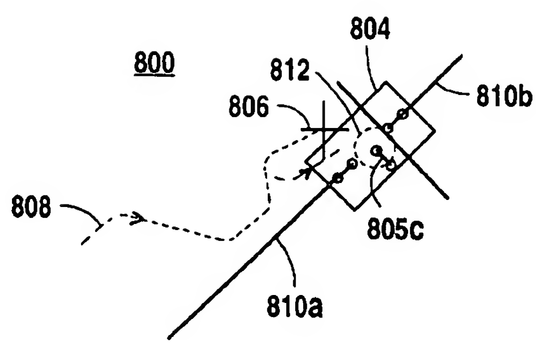


Fig. 8D

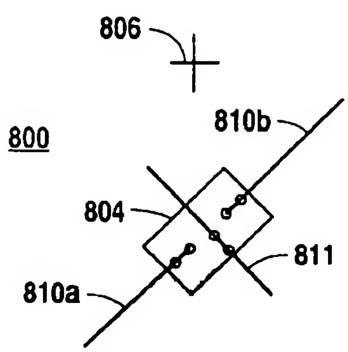


Fig. 8E

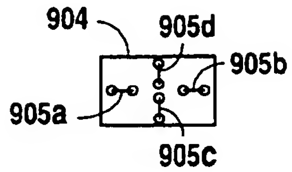


Fig. 9A

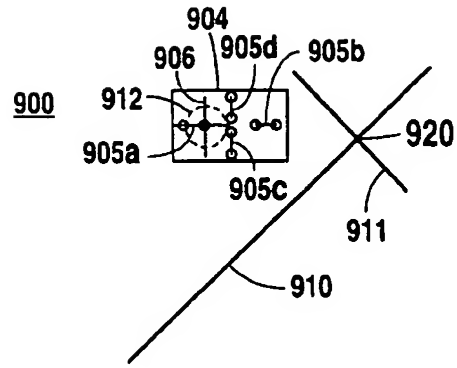


Fig. 9B

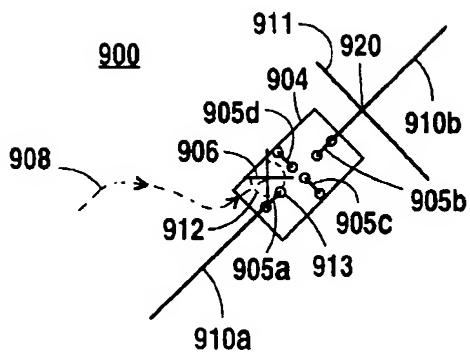


Fig. 9C

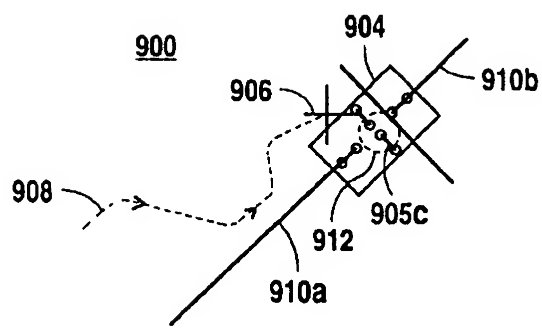


Fig. 9D

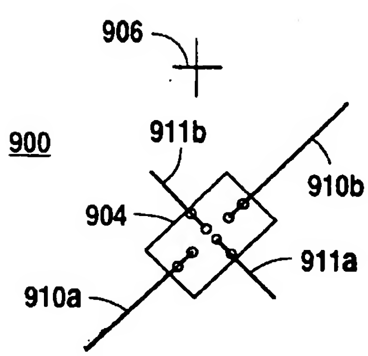


Fig. 9E

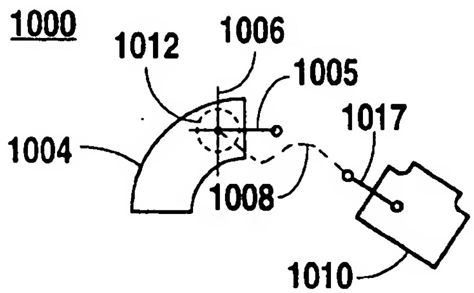


Fig. 10A

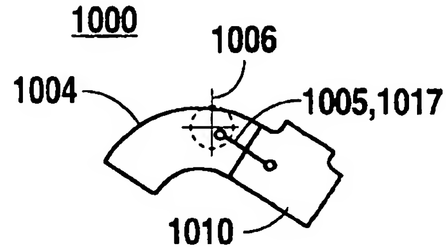


Fig. 10B

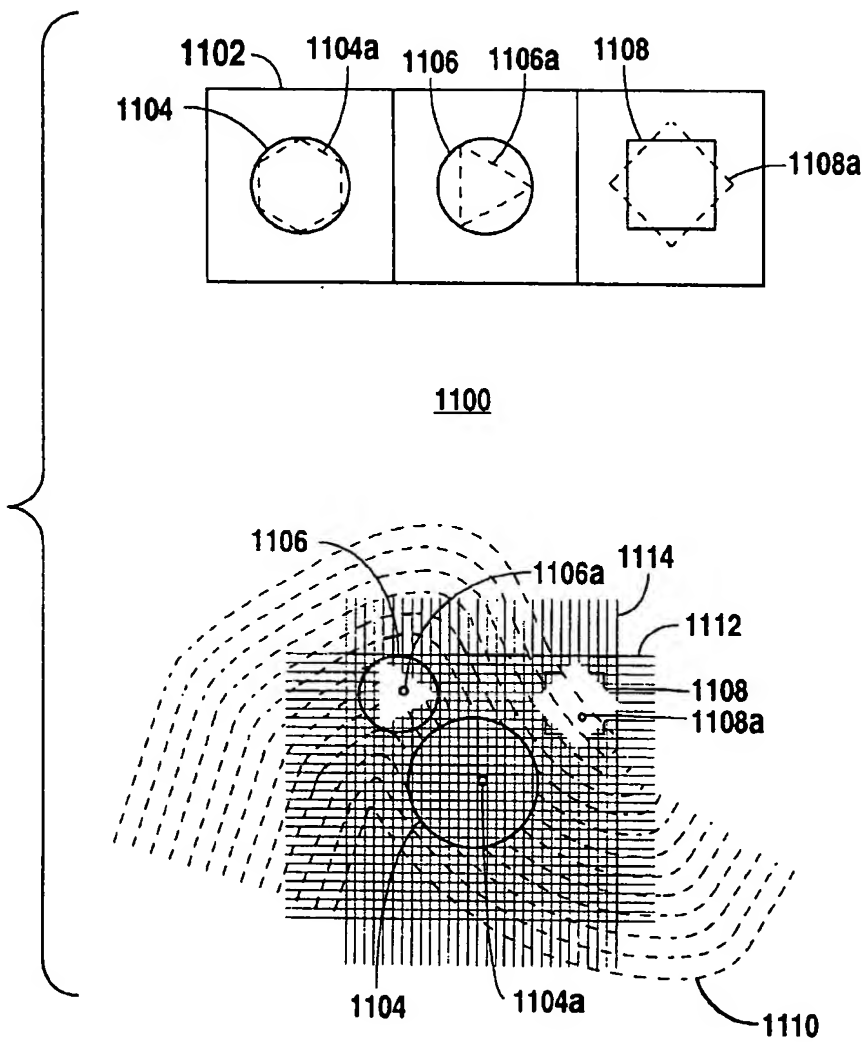


Fig. 11

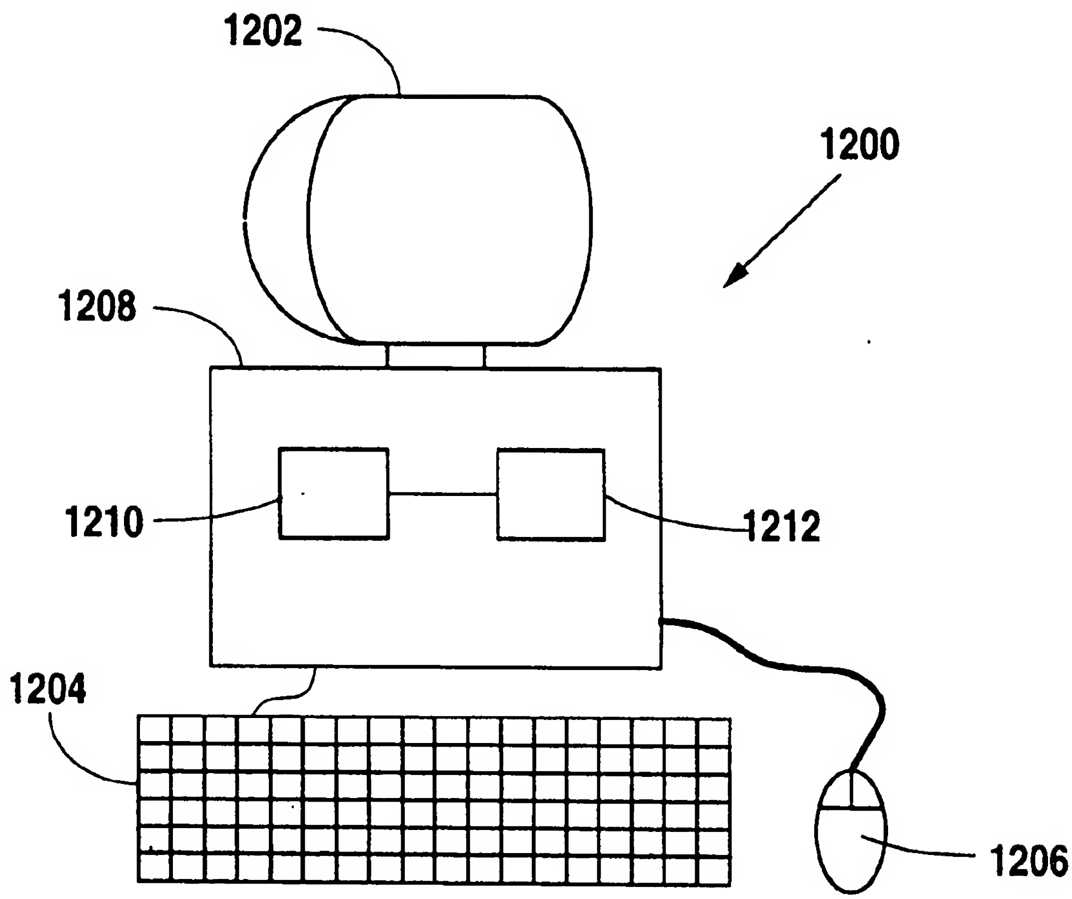


Fig. 12

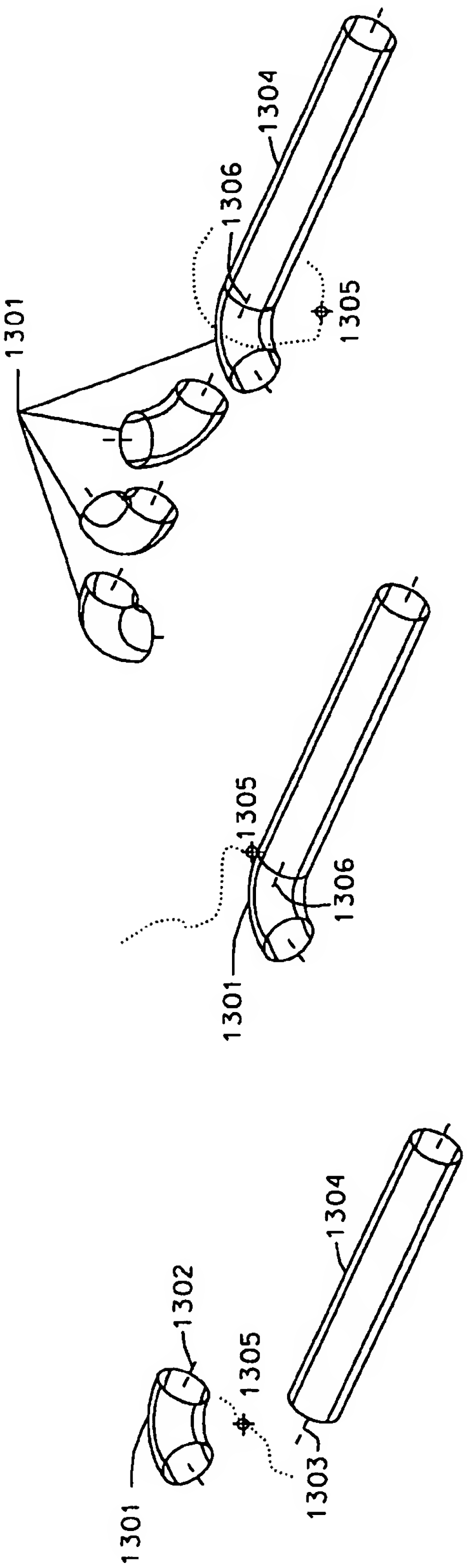


Figure 13C

Figure 13B

Figure 13A

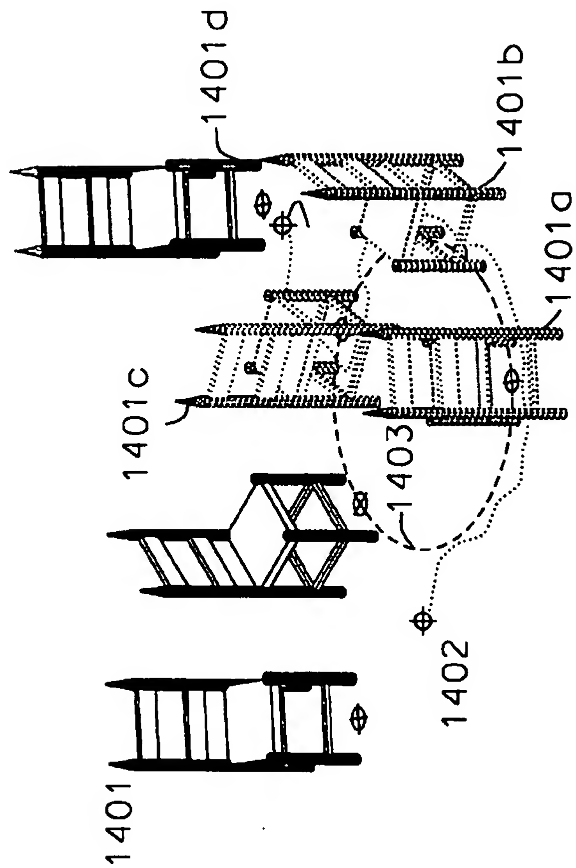


Figure 14B

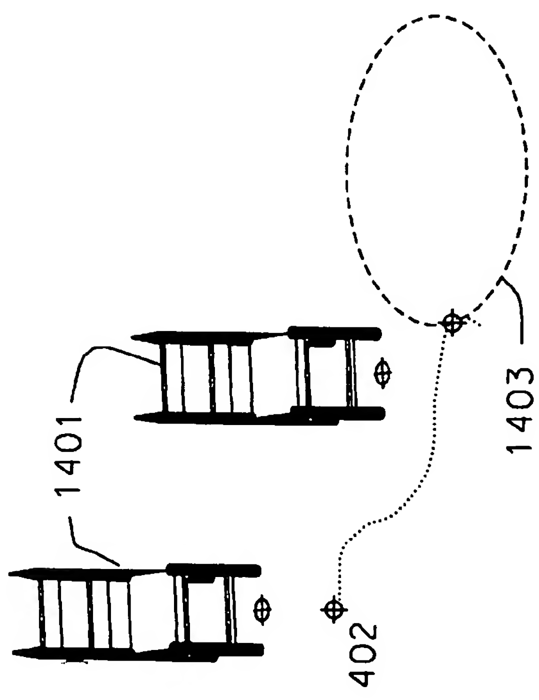


Figure 14A

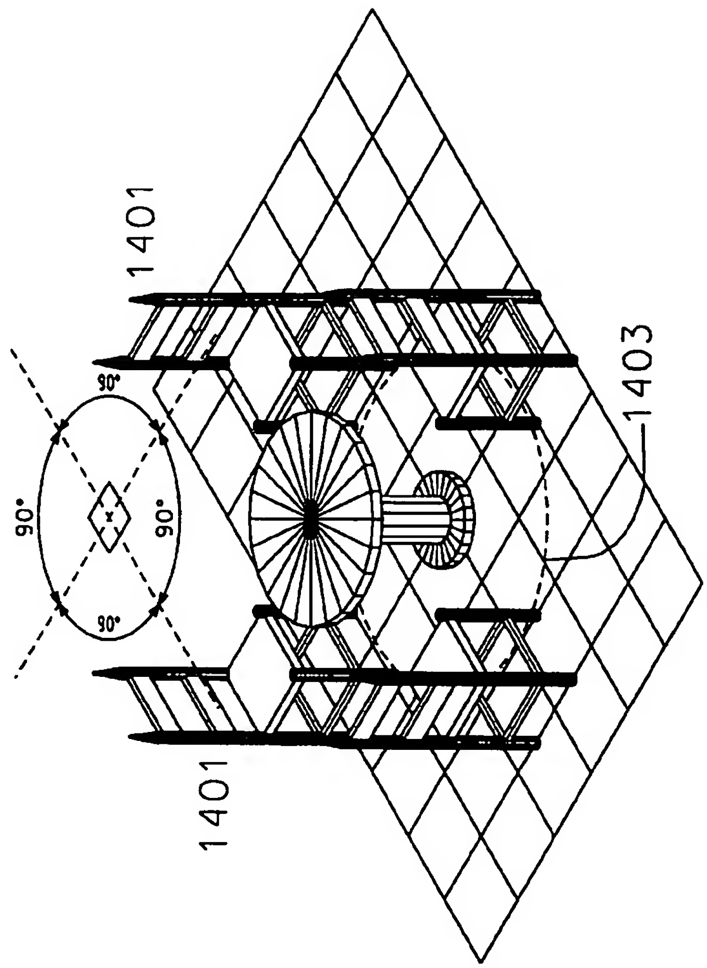


Figure 15B

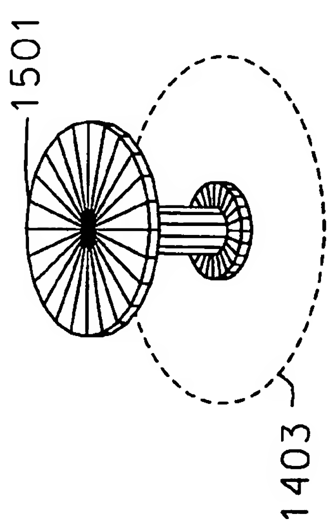


Figure 15A

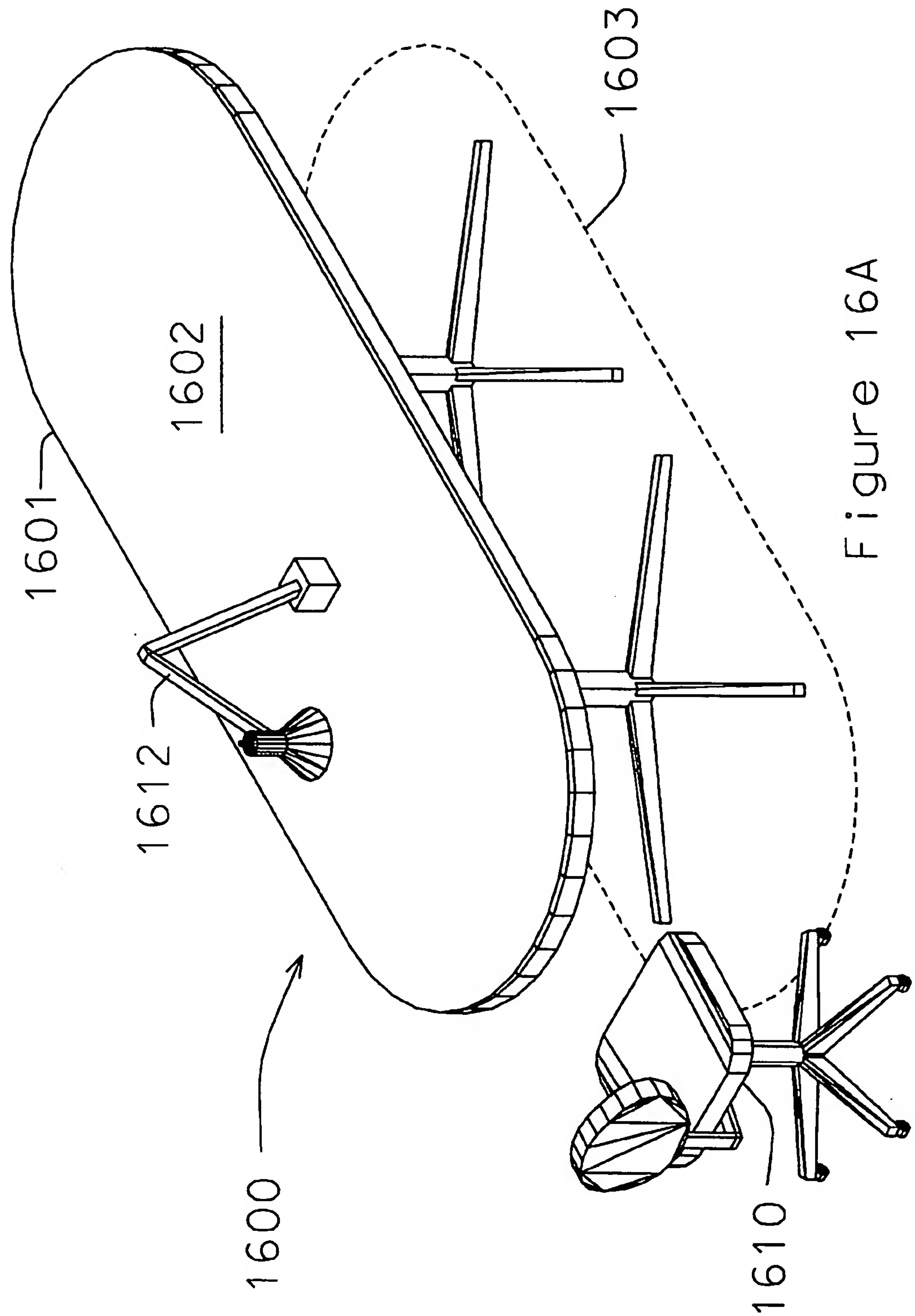


Figure 16A

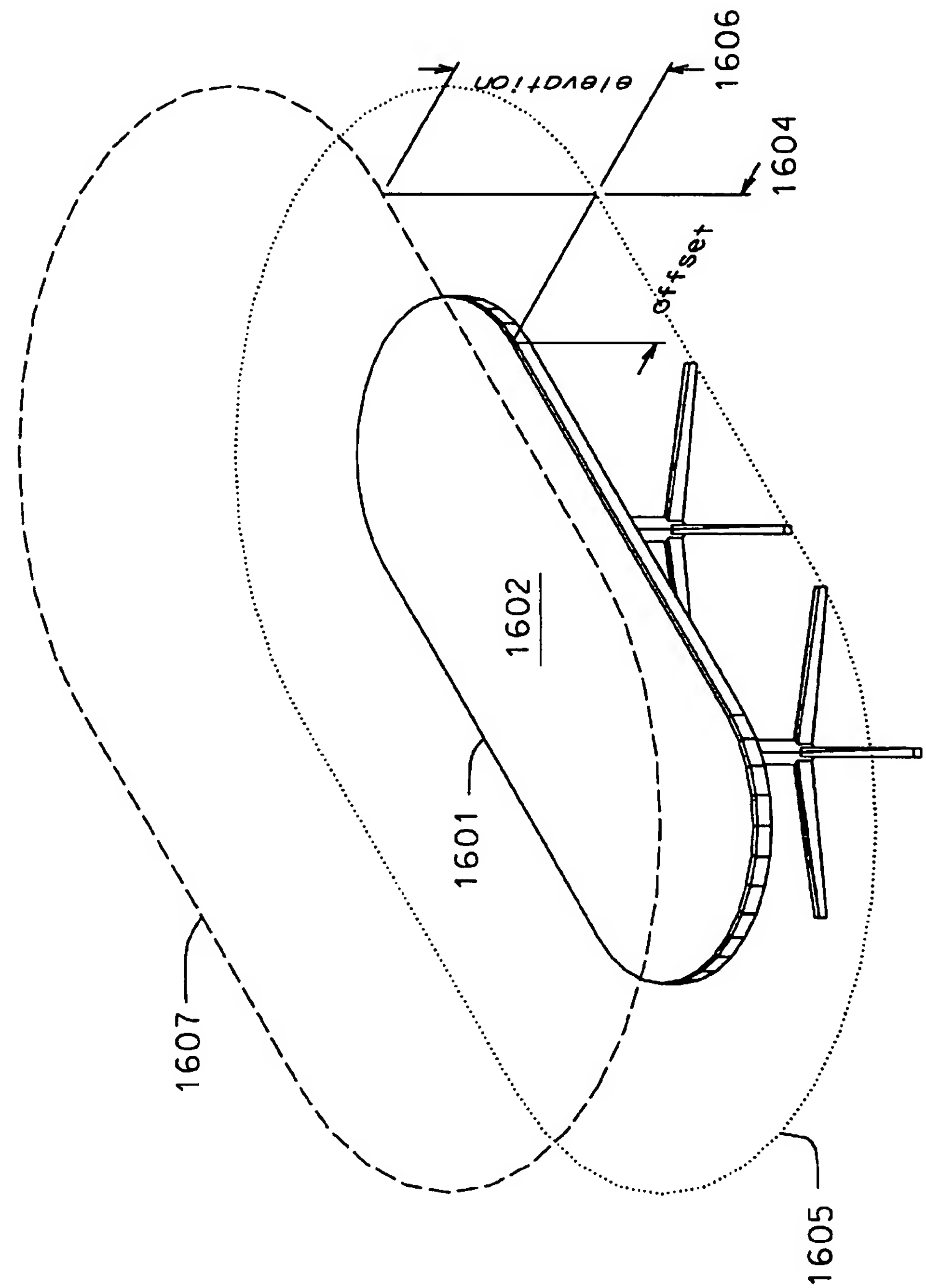


Figure 16B

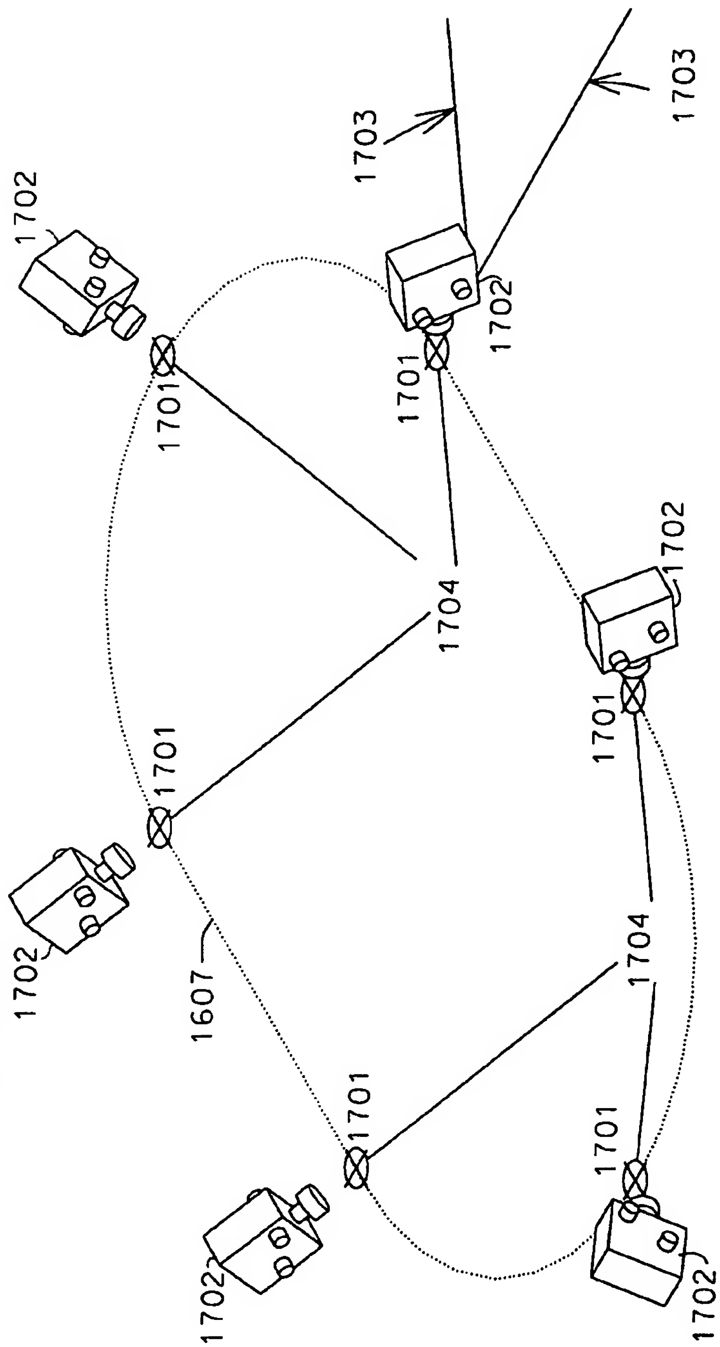


Figure 17

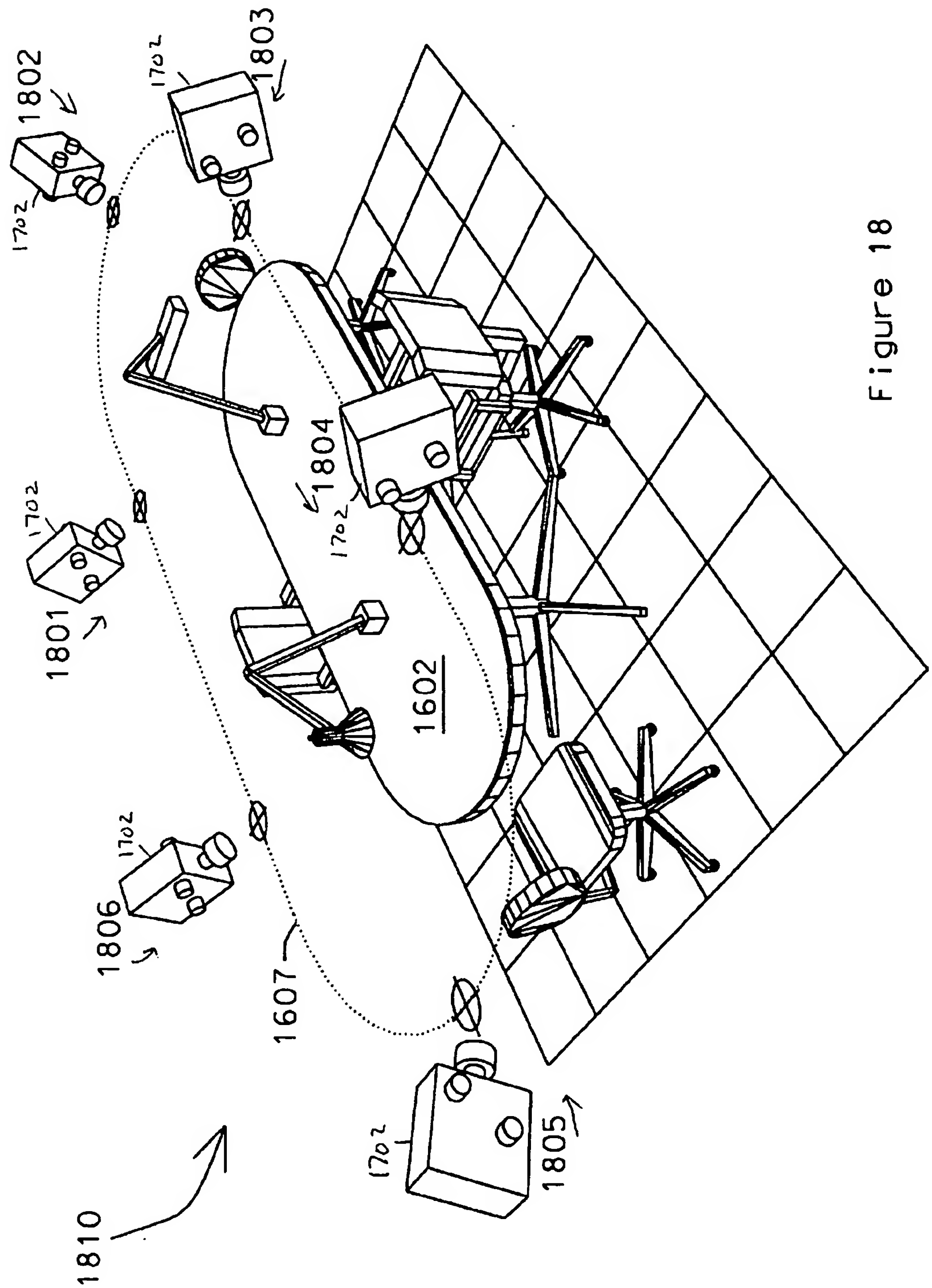

1802
1803
1702
1702
1804
1702
1801
1702
1602
1806
1702
1607
1805
1702
1810

Figure 18

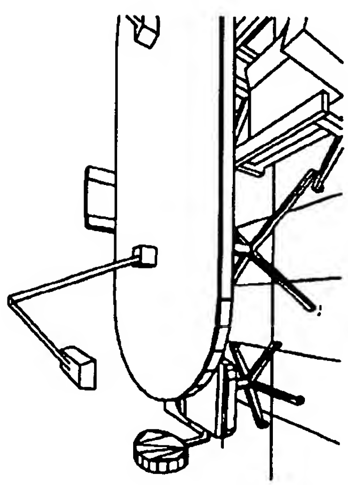


Figure 19A

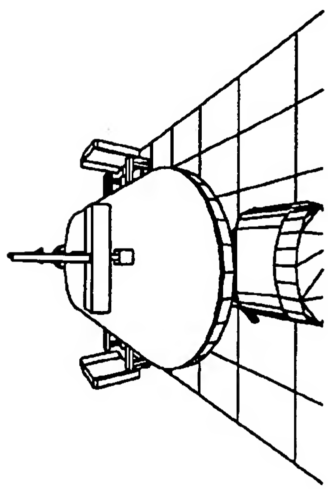


Figure 19B

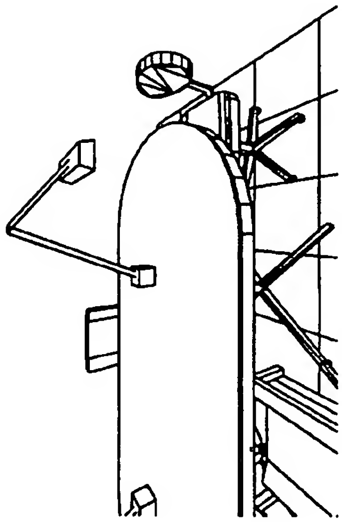


Figure 19C

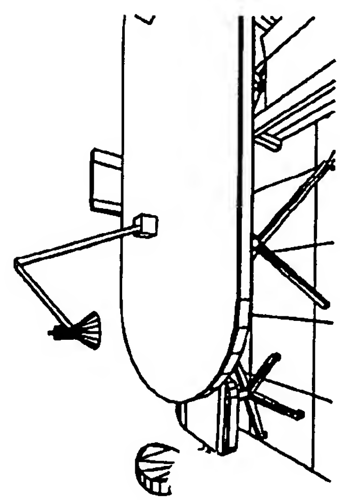


Figure 19D

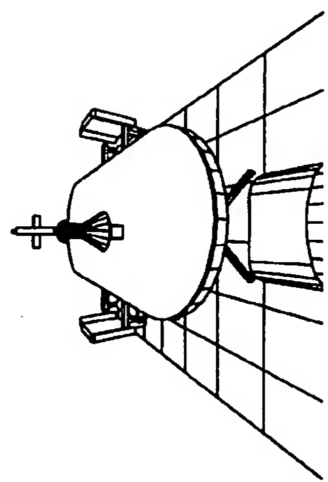


Figure 19E

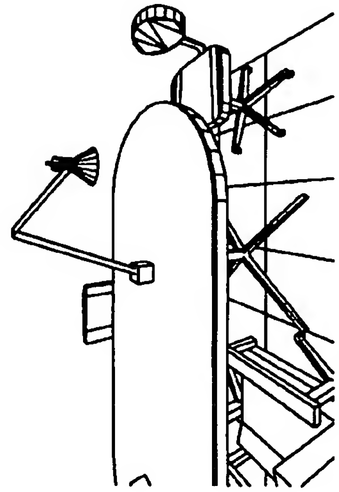


Figure 19F

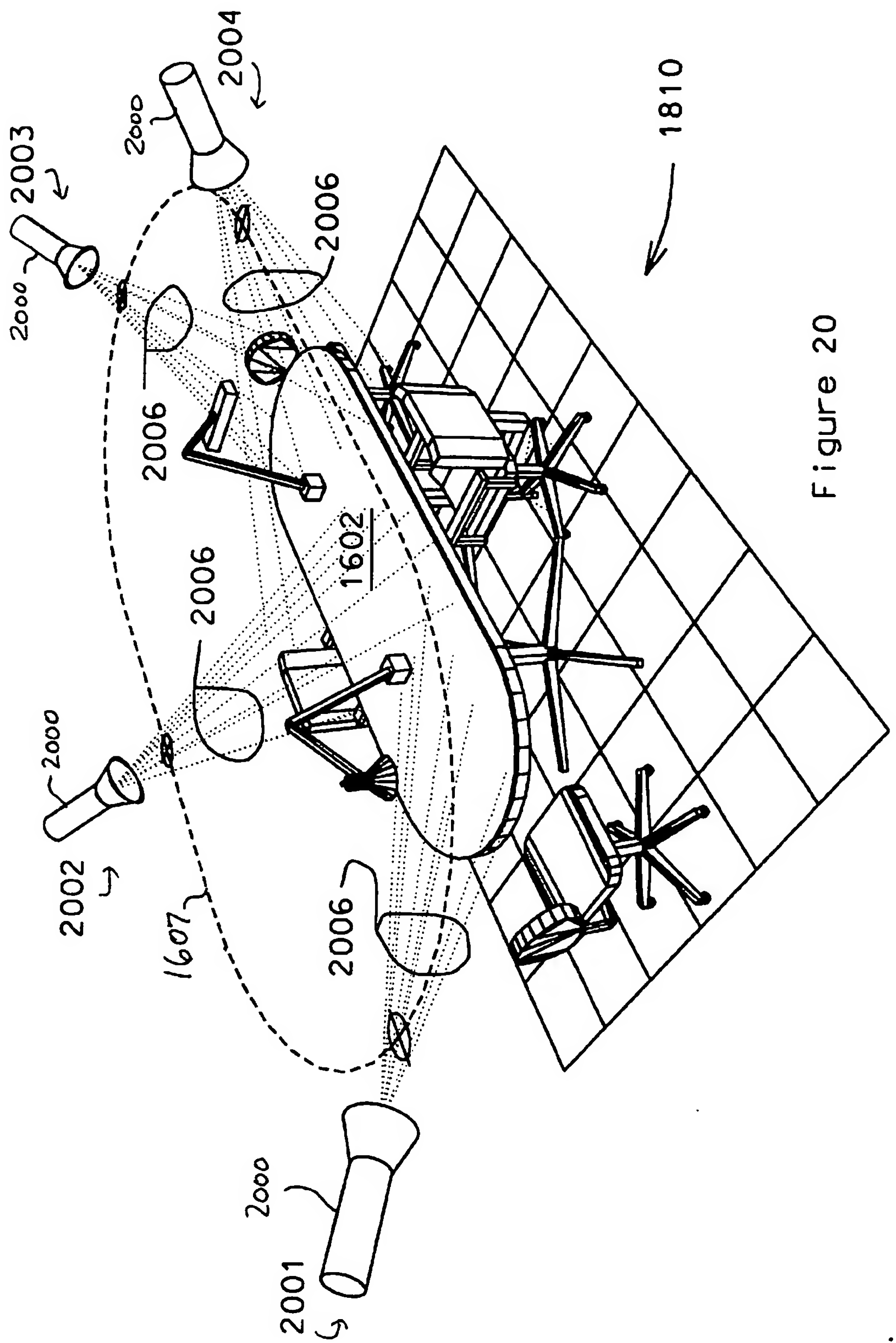

2004
2000
2003
2000
2006
2006
1810
2006
1602
2006
2000
2002
1607
2006
2000
2001

Figure 20

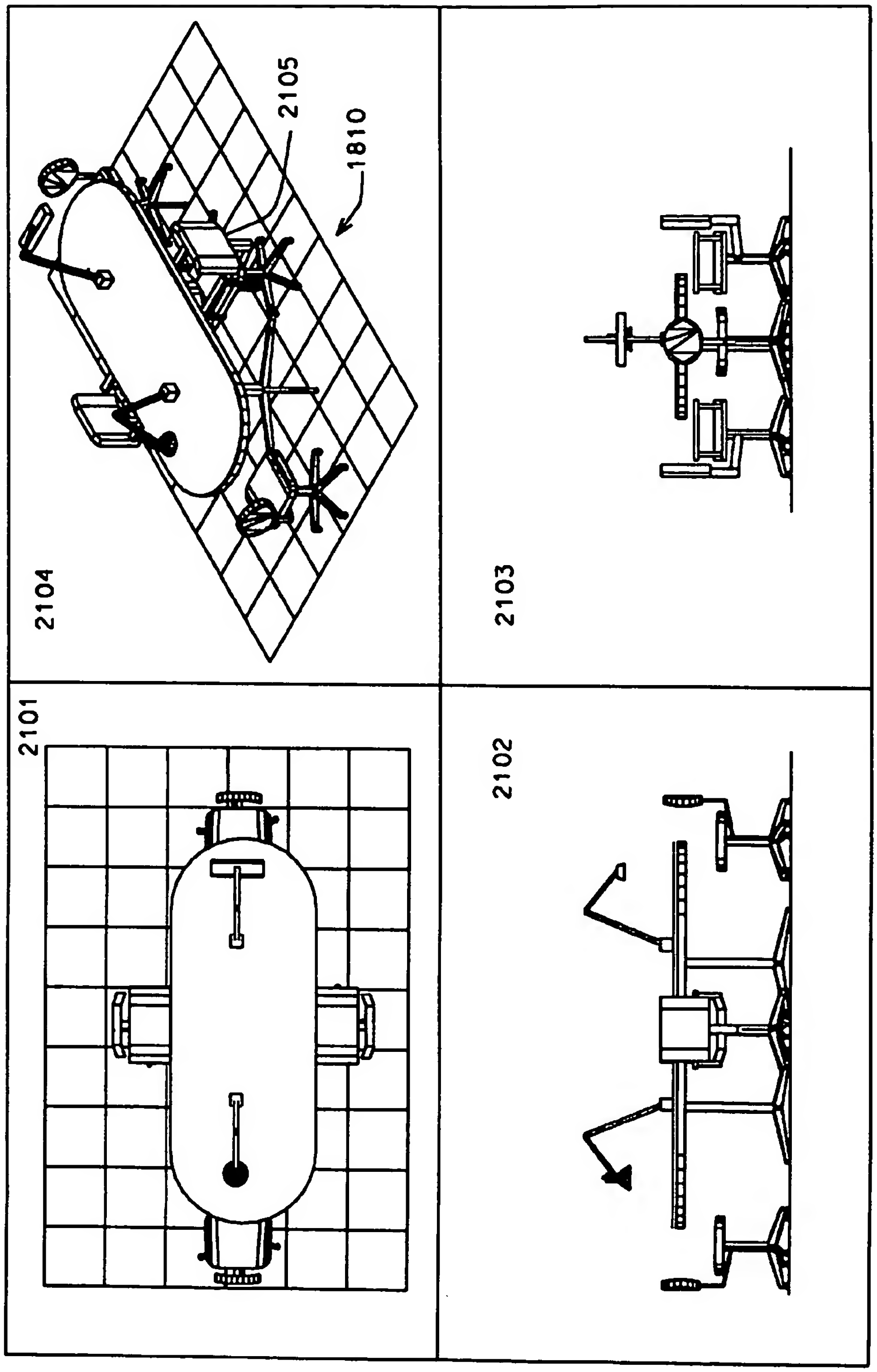


Figure 21

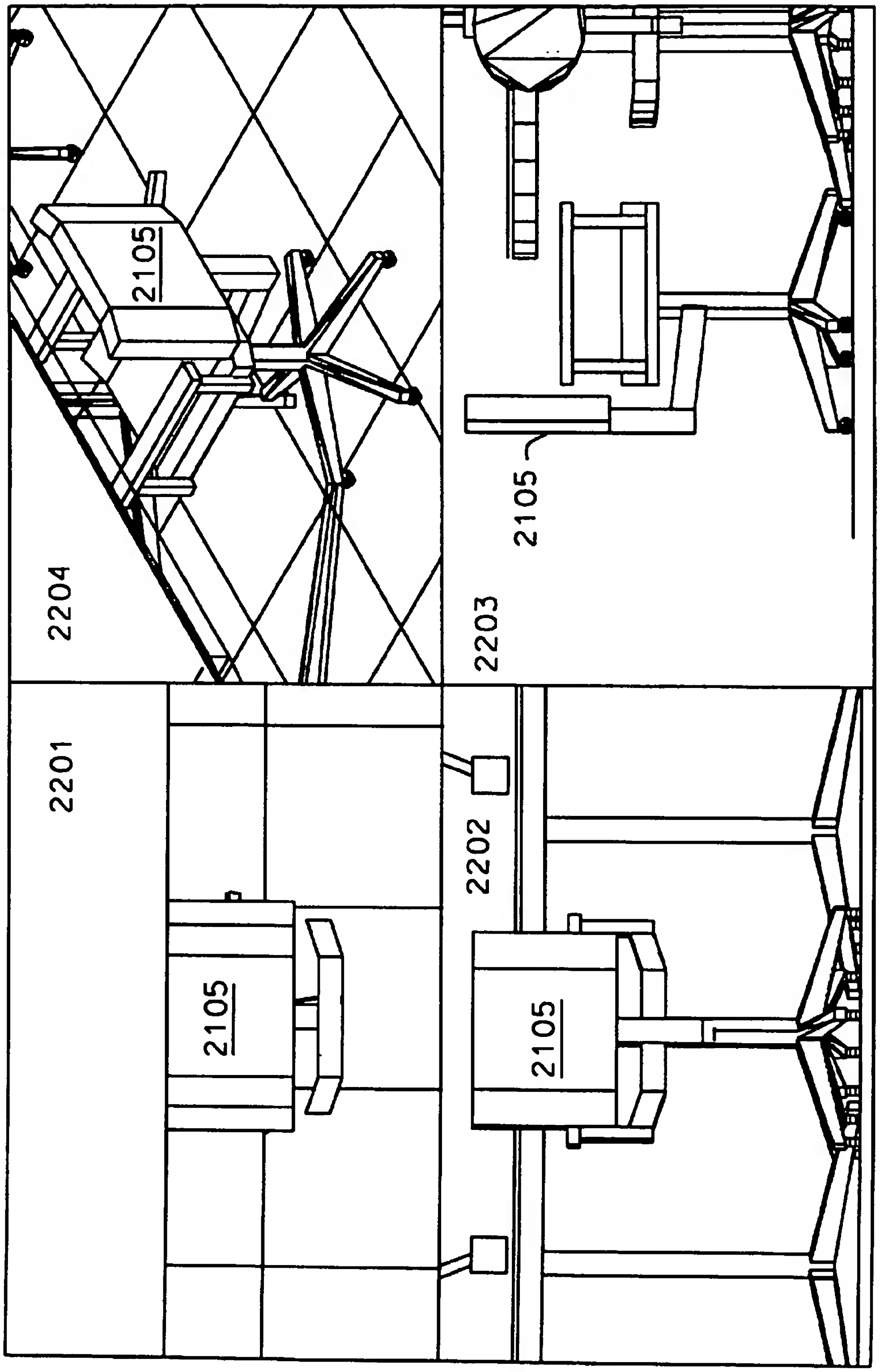


Figure 22

METHOD AND SYSTEM FOR INTERACTIVELY DETERMINING AND DISPLAYING GEOMETRIC RELATIONSHIPS BETWEEN THREE DIMENSIONAL OBJECTS BASED ON PREDETERMINED GEOMETRIC CONSTRAINTS AND POSITION OF AN INPUT DEVICE

CROSS REFERENCE TO RELATED APPLICATIONS

This application is a continuation-in-part of pending U.S. Application entitled "Method And Apparatus For Interactively Manipulating and Displaying Presumptive Relationships Between Graphic Objects" by the same inventor, Ser. No. 08/436,158, filed May 8, 1995, which will issue on Nov. 5, 1996 as U.S. Pat. No. 5,572,639.

FIELD OF THE INVENTION

The present invention relates to computer aided design and drafting systems, and more particularly to interactively determining and displaying geometric relationships between three dimensional objects based on predetermined geometric constraints and position of an input device.

DESCRIPTION OF THE RELATED ART

At the present time, the assembly of three dimensional (3D) objects to realistically depict physical models is based upon the utilization of geometric constraints that roughly correspond to real world physical behavior. Traditional computer-aided drafting (CAD) methods for assembling these types of digital models require that a computer operator indicate where and how 3D objects are to be positioned in digital space. The operator indicates a position and orientation for the 3D graphical objects and the computer subsequently produces the digital representation suggested by operator input. If the resulting representation is not correct, the operator deletes the incorrect graphics from the digital model and attempts to create a new representation that meets the desired criteria.

An operator may press a button on a mouse to provide a "tentative point" to the computer to suggest where an object might be placed in 3D space. Depending upon the type of CAD software used, a second tentative point may be required to fully specify the 3D point of interest. The computer responds by placing a graphic "crosshair" to indicate a precise location nearby the point suggested by the operator. If the point suggested by the operator is close to a key coordinate value from an existing 3D object in the digital design file, the computer places the tentative point at that location and redisplays the graphic object selected in a specified "highlight" color. If the resulting location is desired by the operator, a key is depressed on an input device to accept the tentative point and the specific coordinate values are used one time in an immediately following 3D object input operation. If the coordinate location and associated graphic object determined by the computer is not desired by the operator, the mouse button is pressed again to request a different tentative point.

Some CAD software provides a mode of interaction where the software automatically suggests geometrically interesting points near the cursor for consideration in CAD drafting operations. However, these systems work best in two dimensions where there is little ambiguity regarding the true location of the "interesting point" displayed.

Computer programs exist to create and edit solid models in three dimensions. These solids modeling programs provide interactive tools that allow the merging of individual graphic objects such as a cylinder, block, torus or other objects in order to create a new solid model object definition. Some of these programs offer interactive tools to provide a surface to surface cling mode that constrains the motion of one object's surface to the surface of another. However these programs lack the ability to define a constrained assembly methodology.

Once the user accepts the coordinate location suggested by the computer, a second operation usually follows whereby the 3D object is moved to a correct geometric position, and possibly even rotated about one or more specific coordinate axes to produce the desired orientation. Once again, a tentative point mode of interaction may be used to establish the point of rotation, or else a default coordinate value indicating the origin point of the 3D object may be employed as a rotation point. To specify the exact degree of rotation, another tentative point may be obtained, a specific angle value may be provided by the operator, or else the operator may seek an orientation that is acceptable by moving the mouse, which in turn rotates the 3D object until a satisfactory orientation is obtained. Due to the difficulty of interacting in virtual 3D space with only two dimensional (2D) input devices, such as a mouse or the like, and output devices, such as a video monitor or the like, an operator may have to construct temporary geometric elements to fully constrain placement and orientation. Once the 3D object is in place, the temporary construction geometry is deleted from the 3D model.

To insure that the 3D model has been correctly assembled, the CAD operator desires to visualize the model from different perspective points. In present art, this is accomplished by indicating a point of interest in 3D space using one of the previously described coordinate specification techniques, setting a view distance from the point, specifying the axis of rotation for either the view screen or the virtual world, then either providing an explicit angle value or else interactively changing the viewer's perspective point by moving the mouse or other input device. In general, this method of view manipulation is difficult to master as it requires advance knowledge of how the graphic display system is going to respond to modifications to the underlying mathematical transformation representing 3D space on a 2D video screen.

It is very difficult to create 3D designs using 2D tools, such as a mouse and a computer screen. Consider, for example, the difficulty of placing a 3D chair object at a proper location around a 3D table object. The operator typically uses one perspective, such as a top perspective, to place the chair object to a position that appears to be located adjacent the table object. After placing the chair, however, the operator changes perspective to a side view and discovers that the chair object was placed above the table object and not properly on the floor. The operator must then move the chair object down to the floor. Further, the operator may have to reposition the chair object one or more times to place the chair object at a desired location with respect to the table object. Even if more than one view is provided at a time, the operator is typically constrained to work in one view at a time, and thus must typically manipulate the object in several views before the proper geometric relationship is achieved.

It is an objective of the present invention to more rapidly produce computerized representations of 3D models that conform to predefined specifications for appearance, content and relationships among the graphic objects that are assembled to form the design.

It is a further objective of the present invention to eliminate the duty on the part of the computer operator of providing the correct position and orientation of graphic objects to assemble a valid 3D model of a design or system through a rule-based database to verify the juxtaposition of 3D objects within the intended context of the design.

It is still a further objective of the present invention that the behavior of the graphics objects be constrained by a set of geometric specifications that are constructed in advance of digital data input operations and are encoded in the definition of the 3D objects.

It is still a further objective of the present invention that the position, orientation and projection of the physical model displayed on the screen can be easily altered through the intelligent manipulation of perspective points, viewing distances and rotations of the view without an operator having to master complex, multiple step commands for view manipulation.

It is still a further objective of the present invention that external procedures for the verification of 3D object relationships can occur during digital data input operations to avert the creation of invalid representations of designs.

SUMMARY OF THE INVENTION

A method and system according to the present invention replaces the multiple step mode of 3D coordinate input with a single step assembly methodology defined for each 3D object to be included in the 3D design. In particular, most positions and orientations for 3D objects are tied to the movement of the input device or cursor. A computer system implemented according to the present invention continuously calculates geometric relationships near the cursor for the operator to accept or reject. Predefined rules are maintained to limit the relationships of interest for 3D objects and to perform the geometric computations that provide other related functions such as tangent, offset, parallel, alignment, end point, major vector, divided segment, extended segment, intersection and other specific coordinate locations derived from the graphic objects that comprise a digital 3D design.

In general, there are two classes of intelligent assembly: explicit and inferred. In the case of explicit assembly, there are few options for the positioning of objects, sometimes only one. The case of a grounded electrical plug and outlet is an example of a single orientation. A valve installed in a pipe offers an example of a somewhat less explicit orientation of one object to another. In the case of planning office space, the number of candidate objects and their potential arrangements preclude an explicit definition of assembly. For many of these objects, assembly is inferred from the position of surrounding objects. For the case of a chair, it may sit under a desk, alongside a wall, next to a table or couch, or stand by itself. The position of a chair object into the office design depends upon a number of factors, including aesthetic preference.

Explicit assembly is handled by specifying a few key points or vectors in the geometric definition of the 3D object. Inferred assembly requires more flexibility and variety in the definition of the constraint geometry elements.

In addition to handling 3D objects that represent physical entities, a system and method according to the present invention also handles logical 3D objects such as perspective viewpoints and light sources. An interface is provided to accommodate external rule-based input verification procedures, and the newly input 3D objects inherit specific characteristics of the related 3D object already in the design. A system and method according to the present invention

eliminates much of the labor required for the interactive selection, positioning, orientation and confirmation of 3D objects used in digital models of physical reality.

The present invention provides a method for positioning and displaying 3D objects within the virtually real space of a digital CAD system through geometric affinity between existing and new objects within a 3D design. A system and method according to the present invention uses predefined geometric relationships between 3D objects defined by geometric elements and other related 3D objects in the design and position of an input device, such as a mouse or trackball. The invention also contemplates that logical objects, rather than physical objects, may also be placed into an assembly of 3D objects. Such logical objects include things like the viewer's perspective point, the camera angle used, the location of light sources, etc.

A geometry processing engine written according to the present invention interactively determines the desired positional relationship between graphic 3D objects and automatically generates the correct representation based on geometric constraints and position of the input device. The task of the CAD operator is shifted from instructing the computer on how to position and orient 3D objects to accepting or denying a geometric relationship interactively determined by the computer. Input parameters define constraints that presume the desired layout, so that more often than not the operator accepts the positions of the graphic objects and continues with the next task. This eliminates a great deal of time spent by the operator in constructing geometric relationships between 3D objects.

"Magnetism" is a metaphor that is used for the interactive graphic behavior of the elements. For example, the active position on the screen is controlled by the movements of a pointing device, usually a mouse, trackball, or similar type device. When a 3D object is to be added to a drawing, the object is moved with the cursor and automatically positioned with respect to any appropriate 3D object whenever the cursor partially occludes or passes over an underlying or existing 3D object. As the operator moves the object around in a 3D representation, such as on a computer screen, and as the cursor is moved to partially occlude other graphic elements, the graphic engine dynamically alters the position of the 3D object to agree with the assembly specifications with respect to the currently occluded element. When an occluded element is active, the new 3D object "clings" to the element at a specific location, or else is moved along the extent of the geometric constraint within the occluded element following movements of the cursor (as moved by the input device). To a CAD operator, the 3D object appears to be "magnetically" attracted to the regions surrounding other 3D objects of the drawing. To cancel the magnetic cling behavior, the operator moves the cursor away from the existing 3D object a certain predetermined distance, whereupon the new 3D object behavior reverts to moving with the cursor and automatically aligning with other appropriate 3D objects.

As an option, the 3D object partially occluded by the cursor may be sectioned or cut away a certain distance to provide an exact space to accommodate the new 3D object. The distance can be defined in a variety of ways including the graphic extent of the 3D object, by a closed space definition that is included in the definition of the symbol, and by certain 3D geometric elements included in the definition of the 3D object. The present invention includes the capability to automatically align, orient and cut out space for a 3D object.

A method of interactively determining geometric relationships between 3D objects and displaying the 3D objects

according to the present invention comprises the steps of detecting the position of an input device, moving a selected 3D graphic object relative to a graphic pointing symbol in a 3D representation according to the position of the input device, determining if the selected graphic object is moved to occlude an underlying 3D graphic object in the 3D representation, and positioning and displaying the selected graphic object with respect to the underlying graphic object according to predetermined geometric constraints and the position of the input device.

The method further may include a step of dynamically moving and displaying the selected graphic object according to movement of the input device and the predetermined geometric constraints while the selected graphic object occludes the underlying graphic object. The dynamically moving and displaying steps may further comprise the steps of clinging the selected graphic object to the underlying graphic object, and rotatably moving and displaying the selected graphic object about the underlying graphic object corresponding to movement of the input device. Occlusion may be based on a predefined geometric graphic element associated with the underlying graphic object. Positioning includes orienting and aligning the selected object according to predefined geometric constraints. Such geometric constraints may be defined in association with graphic constraint elements calculated interactively or may be incorporated in the definition of the graphic object.

A method according to the present invention of interactively displaying a 3D design based on geometric constraints and an input device, comprises the steps of displaying a first 3D graphic object having a defined geometric graphic constraint element in a 3D representation, moving and displaying a second 3D graphic object in the 3D representation relative to correspond to the position of the input device, determining if the second graphic object is moved to occlude the geometric graphic constraint element of the first graphic object, and dynamically positioning and displaying the first graphic object into a geometric relationship with the first graphic object according to predetermined geometric constraints defined by the geometric graphic constraint element and position of the input device.

The second graphic object is any type of graphic object including logical graphic objects such as logical cameras or light sources. For a logical camera object, a method according to the present invention may include steps of viewing the first graphic object from the viewpoint represented by the logical camera object, and manipulating the logical camera object by moving the input device to interactively change the display of the first graphic object according to position of the input device. In this manner, the operator may interactively change the viewpoint of a 3D design simply by moving the input device.

A graphics system for interactively determining geometric relationships between three dimensional objects and displaying the three dimensional objects according to the present invention includes a monitor for displaying graphics, a pointing device for indicating location on the monitor, a memory for storing a database of graphic display information and associated geometric constraints, a processor for executing a geometry processing engine based on the database, the geometric constraints and the position of the pointing device for displaying a representation of 3D graphics on the monitor, where the geometry processing engine detects the position of the pointing device, moves a selected 3D graphic object relative to a graphic pointing symbol on the monitor according to the position of the pointing device, determines if the selected graphic object is moved to occlude an underlying 3D graphic object, and if the selected graphic object occludes the underlying 3D graphic object in the 3D graphics, positions and displays the selected graphic object with respect to the underlying graphic object according to predetermined geometric constraints and the position of said pointing device.

A system and method according to the present invention therefore provides a solution to all of the problems listed above by controlling objects to behave in an "assembly aware" manner. Individual objects added to a design behave in a logical manner that is consistent with their intended utilization, installation procedure or other placement constraints. Since 3D objects behave in a logical manner, the level of knowledge required of the CAD operator knowledge decreases to the point where specialized training is not required. For example, much of the geometric instructions the operator provides to the CAD software to achieve the desired 3D assembly conventions is eliminated. This present invention thus opens an entirely new class of CAD applications that can be utilized by anyone familiar with the basics of computer interaction. Particularly, applications with wide spread appeal can be enabled for relative computer novices. This savings of time translate into higher productivity and drastically less training on the part of the CAD operator.

BRIEF DESCRIPTION OF THE DRAWINGS

A better understanding of the present invention can be obtained when the following detailed description of the preferred embodiment is considered in conjunction with the following drawings, in which:

FIG. 1 is a flowchart diagram illustrating operation of a system according to the present invention;

FIG. 2 is a representative computer screen that an operator interacts with using a pointing device to create digital drawings;

FIG. 3A is a graphic diagram illustrating operations performed by a system according to the present invention;

FIG. 3B illustrates an initial cling characteristic of a floating object with an existing, underlying object;

FIG. 3C illustrates a continuing clinging characteristic according to the present invention;

FIGS. 3D–3F illustrate possible behaviors that can be applied to a floating object while it is clinging to an underlying object;

FIGS. 4A–4D illustrate yet further examples of the cling characteristic using a system according to the present invention;

FIG. 5 illustrates how TEXT is handled in context with other graphic objects;

FIGS. 6A–6D, 7A–7D, 8A–8E and 9A–9E illustrate various examples of objects including alignment vectors for aligning the graphic objects and modifying underlying objects;

FIGS. 10A and 10B illustrate alignment of two pipe objects using alignment vectors;

FIG. 11 illustrates the present invention used to implement closed clip region objects for partial deletion of graphic objects in a design;

FIG. 12 is a diagram of a computer system implemented according to the present invention;

FIGS. 13A–13C are three dimensional (3D) graphic diagrams illustrating assembly of two pipe objects;

FIG. 14A is a 3D graphic diagram illustrating a selected 3D object being moved with a graphic pointing symbol

towards an underlying graphic constraint element with predefined geometric constraints;

FIG. 14B is a 3D graphic diagram illustrating the behavior of the selected 3D object of FIG. 14A with the underlying graphic constraint element according to the predefined geometric constraint element constraints when a selected 3D object occludes the graphic constraint element;

FIG. 15A is a 3D graphic diagram illustrating a graphic assembly constraint element that is combined with a 3D object in order to define interactive object placement constraints;

FIG. 15B is a 3D graphic diagram illustrating a 3D design after placing selected 3D objects according to the interactive object placement constraints;

FIG. 16A is a 3D graphic diagram illustrating placement of 3D furniture objects with respect to an underlying 3D object to create a 3D design;

FIG. 16B is a 3D graphic diagram illustrating derivation of additional graphic constraint elements from a primary graphic constraint element for a 3D furniture object;

FIG. 17 is a 3D graphic diagram illustrating how 3D visualization is interactively controlled based upon perspective points derived from an additional graphic constraint element derived in FIG. 16;

FIG. 18 is a 3D graphic diagram illustrating a series of constrained view perspective points as applied to the visualization of an inferred assembly of 3D furniture objects forming a 3D design;

FIGS. 19A–19F are 3D graphic diagrams showing the view from each of the constrained view perspective points of the 3D design of FIG. 18;

FIG. 20 is a 3D graphic diagram illustrating a series of light sources having locations constrained as applied to the visualization of an inferred assembly of 3D furniture objects of the 3D design of FIG. 18;

FIG. 21 depicts four simultaneous views of the 3D design of FIG. 18; and

FIG. 22 illustrates a view operation to achieve multiple sized and centered views of a selected 3D object of interest in the 3D design of FIG. 18.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 12 illustrates a computer system 1200 implemented according to the present invention. The computer system 1200 may be any comparable computer system capable of operating as a computer aided design and drafting (CAD) system, such as an IBM XT, AT or IBM-compatible computer system or the like. The computer system 1200 includes a display device or monitor 1202 for viewing a graphic environment. A keyboard 1204 is also provided for inputting text, as well as a pointing device 1206, such as a mouse, trackball or other similar device, for manipulating graphic objects on the screen of the monitor 1202. A main system unit 1208 includes the necessary logic for running software and processing commands as known to those skilled in the art. For example, a processor 1210, such as the 80386, i486, Pentium, etc. by the Intel Corporation (Intel), is coupled to memory 1212 for executing software implemented according to the present invention.

The computer system 1200 is preferably implemented as a CAD system according to the present invention by loading software incorporating a geometry processing engine into the memory 1212 for execution by the processor 1208 for receiving input and commands from the keyboard 1204 and pointing device 1206 and generating a graphic output on the monitor 1202. Graphic parameters and geometric relationships are defined in database files stored in memory. It is noted that alternative computer systems and interfaces are contemplated, such as three-dimensional (3D) holographic displays for improved visual representation of the graphic environment.

Referring now to FIG. 1, a flowchart diagram is shown illustrating operation of a system according to the present invention. The flowchart illustrates that the system is designed to create representations that conform to predefined specifications for the geometric and logical relationships that exist among graphic objects in a computer based drawing representing a design, system or model.

In step 100, the applicable specific geometric relationships such as alignment, offset, etc. are defined for each entity that is represented in one or more drawings. Additionally, any relationships that are based upon associated database attributes are tabulated and encoded. In the next step 102, the graphic objects used as geometric constraint components are created according to specifications for the desired functional behavior. In the next step 104, any additional generic geometric constraints that may apply are determined and tabulated.

In the next step 106, the constraint definitions for each object are created as a collection of digital data that appears in a recognizable form such as a graphic symbol. Each symbol comprises a series of components, some of which are always displayed on a computer screen as the normal graphic representation of the associated object, some components which are not normally displayed on the screen except as an aid to their definition, some logical verification components are tabulated as a digitally encoded rule-based record that is associated with the symbol, and some components are stored as textual specification data that is provided to the control software at the moment the object is activated for inclusion in the design, system or model. The textual data may be any one of several formats, such as ASCII (American Standard Code for Information Interchange) or the like.

In the next step 108, an object is selected for input by the operator using any of several techniques including the selection of a graphic icon from a computer screen (FIG. 2) that represents the object, typing in a keyed command that causes the object to become active, or any other means of indicating to a software program that the desired object is to be added to the drawing using the geometry processing engine.

In the next step 110, the object is read into the geometry processing engine and graphically interacts with other objects according to the specifications provided in the symbolic definition and the constraints of any external database attribute or knowledge based verification process. Feedback is provided to the operator to indicate the integrity of the proposed relationships between the new object and existing graphic objects in the digital drawing. Such feedback includes changing the color of the affected graphic objects, providing additional on-screen motions to the affected symbol to indicate a correct or incorrect validation result, or providing unique auditory sounds to indicate a correct or incorrect validation result. In the next step 111, the graphic representations are verified against a rule-based database.

In the next step 112, the object is accepted by the operator as being a correct representation at which point the geometry engine inserts the symbol in context into the graphic representation of the design, system or model, taking into account

all geometric control specifications provided with the symbolic definition. Once the new graphic object is added to the existing digital file, the sequence of operations returns to step 108 and drafting operations continue. In particular, steps 108–112 are repeatedly performed in sequential manner until the operator has added all desired objects, and operation is then completed.

Referring now to FIG. 2, a representative computer screen 200 is shown in the context of interactive CAD software. Steps 100–106 have previously been performed at this point so that the operator interactively selects objects in step 108 and accepts a selected object in step 112 until a 2D design is completed. The operator selects objects with a cursor as known for window environments, although the present invention is not limited to a windows environment. A tool palette 202 is provided containing one or more icons that indicate the graphic objects that are available for processing by the geometry engine. A series of objects 204 that have been previously placed appear on the screen 200, which in this particular case is a series of pipes for a plumbing system. Of course, other types of objects are contemplated, such as engineering designs, electrical schematics, utility systems such as power generation and distribution grids, chemical processes, etc. The objects 204 thus are represented in the underlying design file. An optional control panel 206 is provided to specify any additional geometric functions that are to apply to the symbolic object. The balance of the screen depicts a typical interactive computer aided design environment.

FIG. 3A is a graphic diagram illustrating operations performed by a system according to the present invention. A computer screen 300, which is similar to screen 200, is shown including a tool palette 302 for selecting graphic objects. The operator selects a symbol from the tool palette 302 and activates an object 304 with a cursor 306, where the geometry processing engine performs the activation as described above. The selected object 304 moves or "floats" with the cursor 306 (thus called a floating object) at a particular displacement, rotation and orientation according to predetermined criterion. In the example shown, the floating object 304 maintains zero degree rotation with its origin on the cursor 306.

Once selected, the operator moves a pointing device to move the cursor 306 and the object 304 within the computer screen 300 along any desired path 308, and eventually within proximity of an underlying object 310. The floating object 304 is selected and shown on the computer screen 300 but is not made part of the underlying design file until accepted at a desired location by the operator. The underlying object 310 has already been previously accepted and therefore part of the underlying design file. Throughout this disclosure, an underlying object exists in the underlying design file, but a new or selected object to be placed is not made part of the design file until accepted by the operator.

A predetermined and programmed location tolerance, illustrated with a dotted circle 312 but normally not displayed, identifies a minimum perpendicular distance which determines when the object 304 is close enough to the underlying object 310 to establish an association or graphic relationship. When the designated origin point of the object 304 moves to within the location tolerance 312 with respect to the underlying object 310 or with respect to any other object where a graphic relationship is allowed, the cling mode of interaction is invoked whereby the floating object 304 "jumps" onto the underlying graphics object 310 as though it were magnetically attracted. In FIG. 3A, the origin and cursor 306 are positioned at a distance from the underlying object 3 10 greater than the location tolerance 312, so the object 304 remains floating with or otherwise attached to the cursor 306.

FIG. 3B illustrates the initial cling characteristic of a floating object with an existing, underlying object. In particular, once the object 304 is within the location tolerance of the underlying object 310, the floating object 304 jumps from the cursor 306 to cling to the underlying object 310. In the example shown in FIG. 3B, the jump is the shortest or perpendicular distance where the origin of the object 304 aligns and is coincident with the closest or cling point 313 of the underlying object 310. The cling point 313 is typically displayed on the screen 300 for purposes of visual feedback to the operator, although it may alternatively be transparent or invisible if desired.

FIG. 3C illustrates how the floating object 304 magnetically clings to the underlying object 310 as the cursor 306 is moved in proximity with the underlying object 310. As the pointing device is moved by the operator, the object 304 follows the extent of the underlying object 310 and, if an offset distance, rotation angle, or other geometric specification has been defined, the object 304 assumes a position with respect to the geometric specifications and the active magnetic cling point 313 on the underlying object 310. In the example shown in FIG. 3C, a programmed rejection tolerance, illustrated as a dotted circle 314 about the origin of the object 304, is defined where the object 304 remains clinging to the underlying object 310 while the cursor 306 is within the rejection tolerance. The rejection tolerance is preferably larger than the location tolerance to achieve a hysteresis effect. It is noted that the location and rejection tolerances are different parameters which are toggled so that only one is active at a time. The location tolerance determines when an object clings to an underlying object and the rejection tolerance determines when a clinging object unclings from the underlying object.

The cursor path 308 and the underlying object 310 are extended to illustrate the cling characteristic. The floating object 304 "slides" in alignment with the underlying object 310 as the cursor 306 traverses the path 308. In particular, when the cursor 306 is at the locations 320, 322, 324 and 326 as shown, the floating object 310 assumes the corresponding positions 330, 332, 334 and 336, respectively. It is noted that the cursor 306 remains within the rejection tolerance defined for the floating object 304 for the positions 330, 332, 334 and 336.

If the operator desires to un-cling from the underlying graphic object 310, operator moves the cursor 306 a distance greater than the rejection tolerance away from the underlying object 310 and the floating object 304 jumps away from the underlying object 310 to the cursor 306 as though it were magnetically repelled. This is shown at a location 328 of the cursor 306, where the floating object once again floats with the cursor 306 as shown at the position 328. If there is an additional specification for the logical relationship between the floating object 304 and the underlying object 310, and if that relationship is not valid for the particular case, the floating object 304 does not cling to and is prevented from floating near the underlying object by an algorithm that displaces the floating object's position with respect to the on-screen pointing device. An additional warning such as an auditory beep or visual cue such as a sudden red color change in the floating object 304 is issued by the computer.

FIGS. 3D–3F illustrate possible behaviors that can be applied to the floating object 304 while it is clinging to an underlying object 310. These behaviors are predefined

according to geometric constraints for a given object. FIG. 3D illustrates that the object 304 may be spun about an initial cling point 313 by manipulating the cursor 306 around the cling point 313, in contrast with FIG. 3C showing the object 304 predefined to maintain a zero degree orientation regardless of its location. Further, the object 304 does not slide but sticks to the initial cling point and rotates according to movements of the cursor 306. FIG. 3E shows the object 304 positioned at a specified perpendicular offset 315 from cling point 313 in the direction of the cursor 306 and maintaining a zero degree orientation. Note that the floating object 304 jumps to the opposite side of the underlying object 310, as shown as 304A, when the cursor 306 traverses from one side to the other of the underlying object 310. FIG. 3F shows the object 304 (304A) at a 180 degree rotation of the underlying object 310 at a specified perpendicular offset 315 from cling point 313 in the direction of the cursor 306, again on opposite sides of the underlying object 310. Other variations are possible, of course, including multiple instances of the floating object, such as a mirror image of the floating object at a specified perpendicular offset from cling point in the direction of the cursor 306, etc.

FIGS. 4A–4D illustrate yet further examples of the cling characteristic using a system according to the present invention. In each case, a cursor 406 with a floating object 404 is moved within a screen 400 along a path 408 relative to an underlying object 410 already placed on the screen 400. The object 404 is kept a predefined distance from the underlying object 410 relative to a sliding cling point, which slides along the underlying object 410 following the cursor 406. The floating object 404 flops to the other side of the underlying object 410, as indicated at 404A, when the cursor 406 crosses over the underlying object 410 in a similar manner as described previously. It is noted that only one object is shown at any given time in the example of FIGS. 4A–4D, where the designations 404 and 404A illustrate orientation of the same object on opposite sides of the underlying graphic object 410.

Other graphic relationships define the orientation and rotation of the floating object 404 based on the position of the cursor 406. In FIG. 4A, the object 404 is mirrored about the underlying object 410 when flopped to 404A. In FIG. 4B, the object 404 is mirrored about a perpendicular 415 when flopped to 404A. In FIG. 4C, the object 404 is mirrored with respect to both the perpendicular 415 and the underlying object 410 to 404A. In FIG. 4D, the object 404 maintains a parallel relationship to 404A.

FIG. 5 illustrates how TEXT is handled in context with other graphic objects. Once the related symbolic object 510 has been drawn on a screen 500, a TEXT annotation floats with a cursor 506 while obeying constraints for placement of the TEXT. The TEXT is made to align to the underlying graphic object 510 using specified offsets, parallels and tangencies. In the example shown, the TEXT begins with an initial location tolerance, identified by dashed circle 512 and a larger rejection tolerance as illustrated by a dashed circle 514, both with respect to an origin of the TEXT. At first, the TEXT floats with the cursor 506 until the cursor 506 is within the location tolerance, at which time the TEXT jumps to align parallel and at a perpendicular tangent with respect to the underlying graphic object 510, but separated by a predefined offset 515. While the cursor 506 is moved along a path 508, within the rejection tolerance, the TEXT aligns tangentially with the underlying object 510 at the defined offset 515. This is illustrated at cursor positions 520, 522, 524 and 526. When the cursor 506 crosses over the underlying object 510 at point 530, the TEXT preferably jumps to the opposite side, but maintains an orientation to allow the TEXT to be read in normal upwards fashion. A dotted line 532 illustrates the path that the TEXT follows. Furthermore, a characteristic is defined where the TEXT automatically re-aligns itself at 180 degree increments, which occurs between positions 524 and 526, to maintain upward reading orientation. When the cursor 506 is moved outside the rejection tolerance, the TEXT jumps back to float with the cursor 506 at an origin, and the location tolerance is re-established.

FIGS. 6A–6D, 7A–7D, 8A–8D and 9A–9D illustrate various examples of alignment vectors for inserting and cutting graphic objects. FIG. 6A illustrates an object 604 with a single alignment vector 605 having two points, an origin point 605*a* for geometry calculations and an alignment point 605*b* for establishing orientation and direction of the alignment vector 605 and the object 604. Although the object 604 is shown as a simple rectangle, it can be any object following particular alignment rules, such as pipes, electrical components, etc.

FIG. 6B shows a screen 600 with an underlying object 610 and a floating object 604 floating with a cursor 606 for insertion, where the underlying object 610 is illustrated as a single line segment. The object 604 includes an alignment vector 605 where the cursor 606 preferably aligns with the origin point 605*a*. A location tolerance is predefined and indicated by a circular outline 612 around the cursor 606. The object 604 is moved with the cursor 606 along a path 608 and brought within the location tolerance of the underlying object 610, where the object 604 snaps to and aligns with the underlying object 610, as shown in FIG. 6C. In particular, the origin point 605*a* jumps to a cling point 613 and the object 604 and alignment vector 605 rotate to align so that the second point 605*b* lies on top of the underlying object 610. The object 604 now clings and slides along the underlying object 610 in a similar manner described previously, where a rejection tolerance is usually defined for maintaining cling with movement of the cursor 606.

It is noted that the eventual desired result is to "connect" the object 604 to the underlying object 610 at the origin point 605*a*, thereby affecting the underlying object 610 in the data base as well as graphically, if desired. In the example shown in FIG. 6C, the underlying object 610 is preferably split into two separate line segments 610*a*, 610*b* at the origin point 605*a* of the alignment vector 605. The underlying object 610 is preferably immediately modified during the cling action and dynamically updated as the object 604 is moved along the underlying object 610, where the respective lengths of the line segments 610*a*, 610*b* are modified accordingly. Alternatively, the underlying object 610 is not affected until the object 604 is actually accepted at a desired location.

In FIG. 6D, the operator has accepted an appropriate location of the object 604, where the underlying object 610 is split into two separate vectors 610*a* and 610*b* at the common origin point 605*a*. It is appreciated that the operator had to only select the object 604, move the cursor to within a predetermined proximity of an underlying object 610, and the system automatically aligned the object 604 with respect to the underlying object 610 and further modified the underlying object 610 according to predefined rules. Then the operator simply moves the cursor in proximity of the underlying object 610 to select the desired location, and accept the object 604 and the object 604 is added.

FIG. 7A illustrates an object 704 including a double alignment vector 705 in collinear mode with two spaced

vectors **705a** and **705b**, each including origin points and alignment points for directional purposes in a similar manner as shown in FIG. 6A. The separation between the respective origin points of the alignment vectors **705a** and **705b** defines a cut length for cutting an underlying object. In FIG. 7B, a screen **700** is shown including an object **704** selected for connection to an underlying graphic object **710**, which is another line segment as shown. When the object **704** is moved into proximity with the underlying object **710** as shown in FIG. 7C, the origin point of vector **705a** clings to a cling point **713**, the object **704** and vectors **705a**, **705b** rotate to align with the underlying object **710**, and the underlying object **710** is divided into two separate line segments **710a**, **710b** separated by the predefined cut length. Again, the underlying object **710** is either modified or cut immediately or modified after the object **704** is actually accepted. Again, the floating object **704** clings and slides along the underlying object **710** while the cursor **706** is moved within the predefined proximity or rejection tolerance, continually redefining the location of the cut.

Eventually the operator selects the location of the object **704**, and the object **704** is inserted and the underlying object **710** is appropriately divided as shown in FIG. 7D. As a practical example, if a floating object includes specific definitions of collinear vectors, the geometry engine cuts the underlying linear graphic object and connects the resulting linear segments to the collinear vectors. This has the effect of breaking a line and inserting a device that forms part of the line, such as a fuse on a circuit schematic.

FIG. 8A illustrates an object **804** including double alignment vectors **805a**, **805b** in collinear mode with an additional orthogonal alignment vector **805c**. The collinear vectors **805a**, **805b** are two spaced vectors, where all three vectors include an origin point and an alignment point for directional purposes as described previously. The orthogonal alignment vector **805c** is preferably placed between and orthogonally aligned with the collinear vectors **805a**, **805b** as shown. The separation between the collinear vectors **805a**, **805b** defines a cut length.

In FIG. 8B, the object **804** with the alignment vectors **805a**, **805b** and **805c** is selected for interaction with underlying graphic objects **810** and **811**, where the primary vector **810** orthogonally intersects a secondary vector **811** at a point **820** as shown. is Again, a screen **800** is shown including a cursor **806** for locating the object **804**.

When the object **804** is in proximity of the underlying object **810** as shown in FIG. 8C, the collinear vectors **805a**, **805b** cling, align and cut the underlying primary vector **810** into two separate vector objects **810a**, **810b** separated by the predefined cut length in a similar manner as described previously. The origin point of the vector **805a** has a location tolerance for jumping and clinging with the primary vector **810**. The object **804** clings and slides along the primary vector **810**.

As illustrated in FIG. 8D, the orthogonal alignment vector **805c** also has a separate location tolerance defined for its origin for clinging to the secondary vector **811**. Thus, when the origin point of the orthogonal alignment vector **805c** is within its location tolerance with the secondary vector **811**, the object **804** and alignment vectors **805a**, **805b** and **805c** jump so that the origin and alignment points of the vector **805c** align with the underlying vector **811**. The operator may move the cursor **806** about a rejection tolerance, where the object **804** remains static and aligned with the intersection point **820**.

In FIG. 8E, the operator accepts the result, and the underlying primary segment **810** is divided into two collinear line segments **810a**, **810b** separated by the cut length, where the cut length is divided on either side of the secondary vector **811**. In the example shown, the primary vector **810** is divided equally on either side of the secondary vector **811**, although unequal divisions and non-orthogonal intersections, e.g. isometric, etc. are just as easily achieved as desired.

FIGS. 9A–9E are similar to FIGS. 8A–8E, except illustrating primary **905a**, **905b** and secondary **905c**, **905d** collinear alignment vectors defining two separate cut lengths for the primary **910** and secondary **911** underlying objects, respectively. The primary and secondary vectors **910**, **911** are divided into two portions **910a**, **910b** and **911a**, **911b**, respectively, divided by respective cut lengths, and the object **904** is aligned and places as desired.

FIGS. 10A and 10B illustrate operation of alignment vectors for aligning an underlying T pipe object **1010** and a selected elbow pipe object **1004** using alignment vectors on a screen **1000**. The underlying T pipe object **1004** includes an alignment vector **1005** and the T pipe object **1010** includes an alignment vector **1017**, each with an origin point and an alignment point. The operator selects the elbow pipe object **1004** having a predefined location tolerance about the origin point of the vector **1005**. The elbow pipe object **1004** floats with the cursor **1006**, it is within the location tolerance of the origin point of the alignment vector **1017** of the T pipe object **1010**, where the elbow pipe object **1004** is automatically rotated and positioned so that the respective origin points and alignment points of each of the alignment vectors **1005**, **1017** overlap. In this manner, the two objects **1004** and **1010** are automatically aligned with each other by the system, and the operator need only accept or reject the proposed relationship. In particular, if the operator intended to connect the objects **1004**, **1010** as proposed, the relationship is accepted, and if not, the operator simply moves the elbow pipe object **1004** beyond the rejection tolerance for connection with another object as desired.

It is noted that the particular alignment vectors described herein are for purposes of illustration. Thus, alignment vectors need not be collinear nor orthogonal but may be aligned at any desired orientation and angle.

FIG. 11 illustrates the present invention used to implement objects including clip regions for partial deletion of underlying graphic objects in a design. A palette **1102** is provided on a screen **1100**, where the palette includes three objects **1104**, **1106** and **1108**, each having corresponding clip patterns **1104a**, **1106a**, and **1108a**, respectively. Also provided on the screen **1100** is a set of underlying object symbol patterns, including a pattern of splines **1110**, a horizontal line pattern **1112** and a vertical line pattern **1114** intersecting one another as shown. The operator selects one of the objects **1104**, **1106** and **1108** from the palette **1102**, and the selected object floats with the cursor as the cursor is moved across the screen **1100** by the operator. As the selected object coincides with or covers the patterns **1110**, **1112**, or **1114**, a portion of all or certain ones of the underlying patterns **1110**, **1112** and **1114** that are coincident with the corresponding clip region of the selected object is deleted.

In particular, the clip pattern **1104a** deletes the coincident portion of the pattern of splines **1110**, but otherwise does not affect the horizontal or vertical pattern of lines **1112**, **1114**. The clip pattern **1106a** deletes the coincident portion of all of the patterns **1110**, **1112** and **1114**. The clip pattern **1108a** deletes the coincident portion of the horizontal and vertical line patterns **1112**, **1114**, but does not affect the underlying pattern of splines **1110**. This partial deletion is contrasted

with simple masking capability, where the graphic portion of the object is obscured but the object "remains" in the graphic file. Although the present invention may be used for partial masking, partial deletion involves actually deleting the coincident portion of the underlying graphic objects in a selective mode.

It is noted that the partial deletion may be performed interactively as the selected and floating object is moved across the screen 1100. However, this is computationally intensive and may cause a computer system to slow down considerably. Thus, the object is usually drawn and the underlying deletions are preferably performed upon acceptance of object at a desired location.

An example of objects including clip patterns to partially delete any underlying graphic object elements is TEXT, where it is desired to create "white space" for TEXT annotation. The objects to be deleted are contained in a specification for that type of annotation. In FIG. 5, for example, if the TEXT overlaps certain underlying objects, a portion of the object coincident with the TEXT is deleted. Also, if the definition of the floating object includes a closed shape drawn with specific graphic parameters, the geometry object engine causes the CAD system to partially delete all specified graphic objects that fall within the defined region. This has the effect of "cleaning up" graphic elements that would otherwise appear to be visually merged with the floating object.

FIGS. 13A–13C are three dimensional (3D) graphic diagrams illustrating assembly of a 3D elbow pipe object 1301 with an underlying 3D straight pipe object 1304. The operator selects a 3D object, such as the elbow pipe object 1301, using the pointing device 1206, which causes the geometry processing engine to activate the selected object. The elbow pipe object 1301 moves in conjunction with a cursor 1305, which in turn is moved by the operator using the pointing device 1206. As shown in FIG. 13A, the elbow pipe object 1301 includes a geometric graphic constraint element 1302, which is similar to the alignment vector 1005 of the elbow pipe object 1004, and which is part of the definition of the elbow pipe object 1301. The elbow pipe object 1301 is moved relative to the cursor 1305 to within proximity of an existing graphic constraint element 1303 of the straight pipe object 1304. Again, the graphic constraint element 1303 is part of the definition of the straight pipe object 1304.

As shown in FIG. 13B, when the cursor 1305 is moved to occlude a portion of the graphic constraint element 1303 of the straight pipe object 1304, the cling mode of interaction is invoked by the geometry processing engine so that the elbow pipe object 1301 jumps to the straight pipe object 1304 as though it were magnetically attracted. The new position is determined by the geometry processing engine based on the definitions in the graphic constraint element 1303 and position of the cursor 1305, which cursor position is defined by the position of the pointing device 1206 as manipulated by the operator. It is noted that a designated magnetic origin point of a graphic constraint element, such as the graphic constraint element 1302 of the elbow pipe object 1301, is typically defined to be coincident with the cursor 1305 when moved with the cursor 1305 in float mode. However, the locations of the cursor and a graphic constraint elements during float mode may be defined in any manner as desired. The magnetic origin or cling point jumps away from the cursor and towards the underlying graphic object during cling mode as previously described.

As shown in FIG. 13C, as the cursor 1305 is moved around the end of the straight pipe object 1304 by the operator, the geometry processing engine determines and rotates the 3D elbow pipe object 1301 about a cling vector 1306, which is defined as a collinear vector with the graphic constraint elements 1302, 1303 during cling mode. The graphic constraint elements 1302, 1303 are defined to enable the elbow pipe object 1301 to rotate about the cling vector 1306 at any angle, or to rotate only to one or more predetermined angle orientations. For example, as shown in FIG. 13C, four predetermined orientations for the elbow pipe object 1301 with respect to the straight pipe object 1304 are shown as a result of a separate rotational constraint of 90 degrees. In this manner, the geometric relationship between the elbow pipe object 1301 and the straight pipe object 1304 is defined based on the graphic constraint elements 1302, 1303 during cling mode and the position of the pointing device 1206. The geometry processing engine dynamically and interactively updates the 3D display according to the defined geometric relationship.

It is noted that if there is an additional specification for the logical relationship between the new object and the underlying graphic object, and if that relationship is not valid for the particular case, then the new object does not cling to the underlying object, and is prevented from being near the underlying object by displacing the new object's position with respect to the cursor. An additional notice to the user such as an auditory beep or visual cue such as a sudden red color change in the new 3D object may be issued by the computer.

FIG. 14A is a 3D graphic diagram illustrating a selected 3D chair object 1401 being moved with a cursor 1402 towards an underlying circular graphic constraint element 1403 having predefined geometric constraints. As before, the operator selects the 3D chair object 1401 and moves it freely through 3D model space in conjunction with the motion of the cursor 1402, which is moved according to position and movement of the pointing device 1206. The position and orientation of the 3D chair object 1401 is defined only with respect to the cursor 1402 while floating in 3D space and before occluding any underlying graphic objects.

The circular graphic constraint element 1403 is shown in isometric projection, which is predefined as a graphic constraint element by it's color, pattern, weight, layer, class or other graphic parameters. The circular graphic constraint element 1403 is defined alone and is not necessarily part of any underlying graphic object in this case. However, the circular graphic constraint element 1403 is used to represent a real or physical object, such as, for example, a surface for placement of certain 3D objects, such as the 3D chair object 1401.

When the cursor 1402, coincident with a designated magnetic origin point, or cling point, of the selected chair object 1401, is moved to occlude a portion of the circular graphic constraint element 1403 as shown in FIG. 14B, the chair object 1401 is repositioned and displayed by the geometry processing engine into an appropriate position and geometric relationship with the underlying circular graphic constraint element 1403. In particular, the chair object 1401 jumps onto the underlying circular graphic constraint element 1403 as though it were magnetically attracted, and clings to the circular graphic constraint element 1403 while the cursor 1402 is moved in proximity thereto. Thus, as the cursor 1402 is moved by the operator, the cling point of the chair object 1401 follows the extent of the circle defined by the circular graphic constraint element 1403 according to particular geometric specifications.

If an offset distance, rotation angle, or other geometric specification has been defined, the chair object 1401 is

oriented with respect to the geometric specifications and the active cling point. For example, if the circular graphic constraint element **1403** represents a floor surface, the chair object **1401** is oriented to stand on the floor.

FIG. **14B** illustrates various positions and orientations that result as the cursor **1402** is moved about the extent of the circular graphic constraint element **1403**. As the chair object **1401** is moved by moving the cursor **1402** outside the circular graphic constraint element **1403**, it is manipulated into a tangential and perpendicular position with respect to the cling point and the circular graphic constraint element **1403**. Although orientation modifications shown occur with respect to the vertical axis, any 3D vector could be employed as an axis of rotation.

The geometric constraint may be defined to keep the chair object **1401** outside the circle of the circular graphic constraint element **1403** as shown at positions **1401***a* and **1401***b*. Alternatively, the geometric constraint may be defined in such a manner so that if the cursor **1402** is moved to a position within the circular graphic constraint element **1403**, the chair object **1401** assumes an orientation of 180 degrees rotation from the prior orientation, as shown at position **1401***c*.

Position **1401***d* illustrates how the operator causes the 3D object to un-cling from the underlying graphic object. In particular, the cursor **1402** is moved by the operator a specified distance away from the circular graphic constraint element **1403**, the chair object **1401** is displayed to jump away from the underlying circular graphic constraint element **1403** and back to the cursor **1402** as though it were magnetically repelled, and it resumes its float behavior.

FIG. **15A** is a 3D graphic diagram illustrating a geometric graphic constraint element, such as the circular graphic constraint element **1403**, that is combined with a 3D object, such as a table object **1501**, in order to define interactive object placement constraints. Geometric graphic constraint elements are usually defined or calculated according to an interesting or important aspect of a graphic object. The graphic constraint elements are either interactively created and automatically calculated based on a primary constraint of a graphic object, or are defined as part of the definition of the object. For example, the circular graphic constraint element **1403** shown in FIG. **15A** is either calculated based on other constraints of the table object **1501**, or is defined as part of the graphic definition of the circular table object **1501** itself. The table object **1501** is circular and shown as centered within, and concentric with, the circular graphic constraint element **1403**, where the circular graphic constraint element **1403** represents both the outline around the table object **1501**, and the planar floor surface upon which the table object **1501** is placed.

In this case, one or more chair objects **1401** are placed around the table object **1501** to rest on the planar floor surface and oriented to face the table object **1501**. As before, each chair object **1401** is selected and placed by the operator using the pointing device **1206**, where the geometry processing engine interactively and dynamically updates the display and geometric relationships based on the predetermined geometric constraints and position of the cursor as determined from position of the pointing device **1206**. The geometric constraints of the graphic constraint element **1403** are defined as desired, such as allowing the chair objects **1401** to be placed at any angle about the table object **1501**, or may be defined to allow the chair object **1401** to be placed only at particular angles. For example, FIG. **15B** is a 3D design illustrating a geometric constraint where the table objects **1401** are allowed only at 90 degree displacements from each other around the table object **1501**. In all cases, however, each chair object is automatically placed to properly rest on the planar floor surface as defined by the graphic constraint element **1403**.

FIG. **16A** is a 3D graphic diagram illustrating placement of 3D furniture objects with respect to an underlying 3D object to create a 3D design **1600**. In particular, a first graphic constraint element **1601** is defined about the edge of the table top of a 3D table object **1602**. A second graphic constraint element **1603** is defined having the same shape as the first graphic constraint element **1601**, but located at an offset below the edge of the table top of the 3D table object **1602** to represent a floor surface upon which the table object **1602** is placed. Both the first and second graphic constraint elements **1601**, **1603** are made part of the definition of the table object **1601**. The geometry processing engine uses the first and second graphic constraint elements **1601**, **1603** to place and display objects.

As shown, a 3D chair object **1610** is placed and displayed around the table object **1602** by the geometry processing engine using the second graphic constraint element **1603** and position of the pointing device **1206**, to thereby place the chair object **1610** in an appropriate geometric relationship with the table object **1602** in a similar manner as described above. In this manner, when the chair object **1610** occludes the graphic constraint element **1603**, the chair object **1610** is displayed to jump to a position on the floor and oriented to face the table object **1601**. As the pointing object is moved, the display is interactively updated to display the chair object **1610** correspondingly moving around the table object **1602**. Also, a lamp object **1612** is placed by selecting and moving it to occlude the first graphic constraint element **1601**, where the lamp object **1612** is positioned and displayed to properly rest on the top surface of the table object **1602**. In this case, the geometric constraints for the first graphic constraint element **1601** are defined such that while the lamp object **1612** is within the outline of the first graphic constraint element **1601**, the lamp object **1612** is positioned and displayed on the table surface. In this manner, the operator places the lamp object **1612** at any desired position on the top surface of the table object **1602** simply by moving the pointing device **1206**. The geometry processing engine detects the position of the pointing device **1206** and performs the necessary calculations based on the graphic constraints for positioning and displaying the lamp object **1612**. Additional 3D objects are placed in this manner to continuously create and update a 3D design.

FIG. **16B** is a 3D graphic diagram illustrating the derivation of additional constraint geometry elements from the first graphic constraint element for a 3D furniture object. In particular, a third graphic constraint element **1605** is defined using a fixed offset distance **1604** from each point of the first graphic constraint element **1601** in the horizontal plane, where the third graphic constraint element **1605** is coplanar with the first graphic constraint element **1601**. The third graphic constraint element **1605** is either mathematically defined and dynamically updated based on first graphic constraint element **1601**, or is made part of the definition of the 3D table object **1602**. A fourth graphic constraint element **1607** is also defined using a fixed elevation distance **1606** above the second graphic constraint element **1604**, where the fourth graphic constraint element **1607** is in a second plane parallel with the second graphic constraint element **1604**. Again, the fourth graphic constraint element **1607** is either mathematically calculated based on other graphic constraints of the 3D table object **1602** or made part of the definition of the table object **1602**.

The first, second, third and fourth graphic constraint elements **1601**, **1603**, **1605** and **1607** are shown as planar elements. However, geometric constraint elements may be defined using linear, planar, 3D, etc. graphic elements as desired.

FIG. **17** is a 3D graphic diagram illustrating how 3D visualization is interactively controlled based upon perspective points derived from the fourth graphic constraint element **1607** of the 3D table object **1602** of FIG. **16**. In particular, one or more logical camera objects **1702** are selected and placed along the length of the graphic constraint element **1607** in a similar interactive manner as described previously. The logical camera object **1702** is moved with a cursor until a defined cling point clings to the graphic constraint element **1607**, where each cling point becomes a view perspective tangent point **1701**.

While clinging to the graphic constraint element **1607**, each logical camera object **1702** is orthogonally aligned with the graphic constraint element **1607**. As shown, each logical camera object **1702** represents its associated view perspective tangent points **1701**, where each logical camera object **1702** initially points inwards with respect towards the graphic constraint element **1607**. In the example shown, an adjustable angle value **1703** is applied to rotate each logical camera object **1702** parallel to the graphic constraint element **1607** in order to orient the view onto a region of interest **1704**. Each logical camera object **1702** may be aligned individually, or any two or more of the logical camera objects **1702** may be selected and aligned simultaneously. The region of interest **1704** is preferably associated with the 3D table object **1602**, so that each logical camera object **1702** is aligned to point towards the 3D table object **1602**.

FIG. **18** is a 3D graphic diagram illustrating a series of logical camera object positions **1801**, **1802**, **1803**, **1804**, **1805** and **1806** of a camera object **1702**, each position representing a constrained view perspective point as applied to the visualization of an inferred assembly of 3D furniture objects including the 3D table object **1602**. Each of the 3D furniture objects were placed about the table object **1602** forming an inferred assembly 3D design **1810**. Note that a single logical camera object **1702** is interactively moved by the operator moving the pointing device **1206**, so that the logical camera object interactively slides along the line of the graphic constraint element **1607** to any position, where the positions **1801**–**1806** are shown for purposes of illustration. Alternatively, the positions **1801**, **1802**, **1803**, **1804**, **1805** and **1806** may represent a predetermined number of viewpoints to provide exact divisions of intermediate view points.

FIGS. **19A**–**19F** are 3D graphic diagrams showing the view perspective of the 3D design **1810** of FIG. **18** from each of the constrained view perspective positions **1801**–**1806**, respectively. In particular, the operator selects the viewpoint of the logical camera object **1702**, and then moves the pointing device **1206** to interactively change the display based upon the viewpoint of the camera object. The geometry processing engine continuously detects the position of the pointing device **1206**, and continuously and interactively updates the display on the monitor **1202** according to the viewpoint of the logical camera object **1702**. Each of the views shown in FIGS. **19A**–**19F** represents only a snapshot of what a CAD operator sees on the screen of the monitor **1202** as a result of six particular constrained view manipulation positions **1801**–**1806**. Note that the top surface of the 3D table object **1602** remains the focus of each scene throughout all six view perspective points **1801**–**1806**.

In this manner, the operator defines a geometric constraint associated with a graphic object in a 3D graphic design, selects and positions a logical view object such as a logical camera, changes the viewpoint displayed on the monitor **1202** to that of the logical view device, and then moves the pointing device **1206** to interactively view the 3D graphic design at any desired angle.

FIG. **20** is a 3D graphic diagram of the 3D design **1810** of FIG. **18** illustrating placement of a logical light source **2000** at any location as represented by locations **2001**, **2002**, **2003** and **2004**, each location constrained by an applied visualization of the inferred assembly of the 3D furniture objects. The logical light source **2000** was selected and placed in association with the graphic constraint element **1607** in a similar manner as described above for the logical camera object **1702**. Each of the logical light source positions **2001**–**2004** are oriented in accordance with the principles of the present invention, using the same interactive techniques as described previously. The logical light source **2000** is moved and rotated with respect to the graphic constraint element **1607** in a similar manner as described above for the logical camera objects **1702** to a specific positions and specific angle in order to project illumination on the top of the 3D table object **1602**. Light rays are depicted as a conical shaped array of lines **2006** emanating from the logical light source **2000** in each of the positions **2001**–**2004** as an aid to positioning the logical light source with respect to a 3D object of interest. Alternatively, multiple logical light sources **2000** may be placed, one for each of the positions **2001**–**2004**.

It is noted that the geometry processing engine may be implemented with commands to interactively illustrate and update the effect of one or more logical light sources **2000** while being positioned. Such effects would include shadow effects for each of the underlying graphic objects, such as the table object **1602**. Alternatively, a separate post-processing portion of the geometry processing engine is executed on the 3D design **1810** after being assembled to determine and display lighting effects based on selected placement of one or more light sources.

FIG. **21** depicts four simultaneous views **2101**, **2102**, **2103** and **2104** of the 3D design **1810**. The views are used in tandem to assist the CAD operator in obtaining precise 3D coordinate points and for visualizing the exact layout of the 3D design **1810**. The particular views **2101**, **2102**, **2103** and **2104** are drafting standard top, front, right side and isometric views, respectively.

FIG. **22** illustrates a view operation to achieve multiple simultaneous sized and centered views **2201**, **2202**, **2203** and **2204** of a selected 3D object of interest. In particular, the operator has selected a 3D chair object **2105** in the 3D design **1810** of FIG. **21** for view manipulation. Again, the views **2201**–**2204** are drafting standard top, front, right side and isometric views, respectively, of the 3D chair object **2105**. The view size for each of the simultaneous views **2201**–**2204** is set to 110% of the 3D extent of the chair object **2105**. The geometry processing engine caused the computer graphics system to display each view centered on the object **2105**.

It is now appreciated that a CAD system according to the present invention enables interactive manipulation and viewing of selected 3D objects in a 3D representation according to predefined geometric constraints and the position of an input device. A system according to the present invention automatically calculates and displays the correct geometric relationships in an interactive fashion. Thus, the

present invention allows an operator to more rapidly produce accurate 3D designs that conform to predefined specifications for appearance, content and relationships among the graphic objects that convey cognition for the intent of 3D designs. The computer operator is relieved of the duty of learning the correct layout of graphic objects to assemble a valid representation of a 3D design, system or model. In effect, a system according to the present invention is an "expert" CAD system, so that the operator need not be very knowledgeable to interactively create and view 3D designs.

Although the system and method of the present invention has been described in connection with the preferred embodiment, it is not intended to be limited to the specific form set forth herein, but on the contrary, it is intended to cover such alternatives, modifications, and equivalents, as can be reasonably included within the spirit and scope of the invention as defined by the appended claims.

I claim:

1. A method of interactively determining geometric relationships between three dimensional objects and displaying the three dimensional objects, comprising the steps of:

detecting the position of an input device;

moving a selected three dimensional graphic object relative to a graphic pointing symbol in a three dimensional representation according to the position of the input device;

determining if the selected graphic object is moved to occlude an underlying three dimensional graphic object in the three dimensional representation;

if the selected graphic object occludes the underlying three dimensional graphic object in the three dimensional representation, dynamically altering a position of the selected graphic object to agree with assembly specifications with respect to the underlying graphic object according to predetermined geometric constraints and the position of the input device; and

dynamically moving the selected graphic object after having been dynamically altered to agree with the assembly specifications according to movement of the input device and the predetermined geometric constraints while the selected graphic object occludes the underlying graphic object.

2. The method of claim 1, wherein said dynamically moving and displaying steps further comprise the steps of:

clinging the selected graphic object to the underlying graphic object; and

rotatably moving and displaying the selected graphic object about the underlying graphic object corresponding to movement of the input device.

3. The method of claim 1, wherein occlusion is based on a predefined geometric graphic element associated with the underlying graphic object.

4. The method of claim 1, wherein the dynamically altering step includes the step of:

orienting the selected graphic object according to a tangential angle with respect to the underlying graphic object at a cling point.

5. The method of claim 1, wherein the dynamically altering step includes the step of:

positioning the selected graphic object at a predetermined offset relative to a cling point between the selected graphic object and the underlying graphic object.

6. The method of claim 1, wherein the underlying graphic object includes at least one graphic constraint element, the dynamically altering step further comprising the step of:

aligning the selected graphic object with the underlying graphic object according to the graphic constraint element of the underlying graphic object.

7. The method of claim 1, wherein the selected graphic object and the underlying graphic object each have a graphic constraint element, wherein the dynamically altering step comprises the step of:

aligning the selected graphic object with the underlying graphic object by aligning the respective graphic constraint elements.

8. A method of interactively displaying a three dimensional design based on geometric constraints and an input device, comprising the steps of:

displaying a first three dimensional graphic object having a defined geometric graphic constraint element in a three dimensional representation;

moving and displaying a second three dimensional graphic object in the three dimensional representation relative to correspond to the position of the input device;

determining if the second graphic object is moved to occlude the geometric graphic constraint element of the first graphic object;

dynamically altering a position of the second graphic object to agree with assembly specifications with respect to the first graphic object according to predetermined geometric constraints defined by the geometric graphic constraint element and position of the input device; and

dynamically moving the second graphic object after it has been dynamically altered to agree with the assembly specifications according to movement of the input device and the predetermined geometric constraints while the second graphic object occludes the first graphic object.

9. The method of claim 8, wherein the second graphic object is a logical graphic object.

10. The method of claim 9, wherein the second graphic object is a logical camera object representing a viewpoint of the first graphic object.

11. The method of claim 10, further comprising the step of:

manipulating the logical camera object with the input device to obtain a desired viewpoint of the first graphic object.

12. The method of claim 10, further comprising the step of:

viewing the first graphic object from the viewpoint represented by the logical camera object.

13. The method of claim 10, further comprising the steps of:

viewing the first graphic object from the viewpoint represented by the logical camera object; and

manipulating the logical camera object by moving the input device to interactively change the display of the first graphic object according to position of the input device.

14. The method of claim 9, wherein the second graphic object is a logical light source object.

15. The method of claim 14, further comprising the step of:

manipulating the logical light source object with the input device to obtain a desired position of the logical light source.

16. A graphics system for interactively determining geometric relationships between three dimensional objects and displaying the three dimensional objects, comprising:

a monitor for displaying graphics;

a pointing device for indicating location on said monitor

a memory for storing a database of graphic display information and associated geometric constraints;

a processor for executing a geometry processing engine based on said database, said geometric constraints and the position of said pointing device for displaying a representation of three dimensional graphics on said monitor, wherein said geometry processing engine detects the position of said pointing device, moves a selected three dimensional graphic object relative to a graphic pointing symbol on said monitor according to the position of the pointing device, determines if the selected graphic object is moved to occlude an underlying three dimensional graphic object, if the selected graphic object occludes the underlying three dimensional graphic object in said three dimensional graphics, dynamically alters a position of the selected graphic object with respect to the underlying graphic object to agree with assembly specifications according to predetermined geometric constraints and the position of said pointing device, and dynamically moving the selected graphic object after it has been dynamically altered to agree with the assembly specifications according to movement of said pointing device and the predetermined geometric constraints while the selected graphic object occludes the underlying graphic object.

* * * * *

Disclaimer

6,016,147—Brian D. Gantt, Travis County, Texas. METHOD AND SYSTEM FOR INTERACTIVELY DETERMINING AND DISPLAYING GEOMETRIC RELATIONSHIPS BETWEEN THREE DIMENSIONAL OBJECTS BASED ON PREDETERMINED GEOMETRIC CONSTRAINTS AND POSITION OF AN INPUT DEVICE. Patent dated January 18, 2000. Disclaimer filed January 22, 2001, by the assignee, Autodesk, Inc.

The term of the patent shall not extend beyond the expiration date of Pat. No. 5,572,639.

(Official Gazette, May 1, 2001)